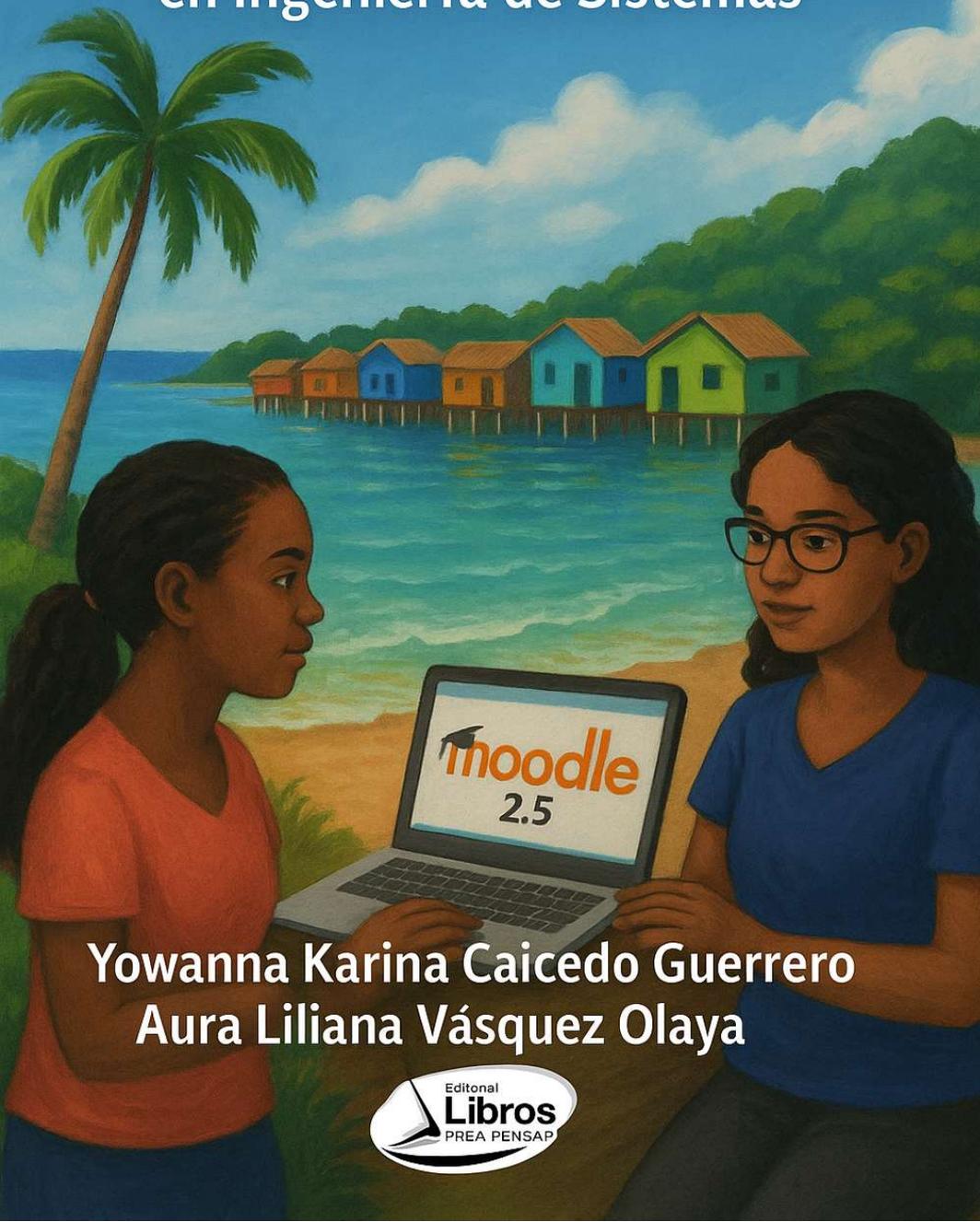
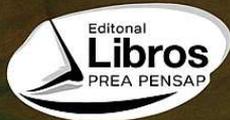


# Moodle 2.5 para fortalecer Fundamentos de Programación en Ingeniería de Sistemas



**Yowanna Karina Caicedo Guerrero**  
**Aura Liliana Vásquez Olaya**





Título: Moodle 2.5 para fortalecer Fundamentos de Programación en Ingeniería de Sistemas

Autoras: Yowanna Karina Caicedo Guerrero & Aura Liliana Vásquez Olaya

Edición: Leonardo Valencia Echeverry

Diagramación: Leonardo Valencia Echeverry

© Yowanna Karina Caicedo Guerrero & Aura Liliana Vásquez Olaya

© EDITORIAL: LIBROS PARA PENSAR

Primera Edición 2025

ISBN: 978-628-01-8745-7

No se permite la reproducción total o parcial de este libro, ni su incorporación a un sistema informático, ni su transmisión en cualquier medio, sea este electrónico, mecánico, por fotocopia u otro método, sin el permiso previo y por escrito del autor.

Hecho en Colombia

Printed in Colombia

Queda hecho el Depósito Legal





## Resumen

La enseñanza de la programación en la educación superior enfrenta desafíos persistentes ligados a modelos pedagógicos desactualizados y a una baja apropiación de las tecnologías educativas. Este ensayo defiende que Moodle 2.5, cuando se emplea con criterio didáctico y compromiso contextual, puede ser una herramienta poderosa para transformar el aprendizaje de programación.

A través del estudio de caso realizado en la Universidad del Pacífico, se demuestra que una integración tecnopedagógica bien diseñada contribuye a mejorar la comprensión algorítmica, reducir la deserción y democratizar el acceso al conocimiento en territorios excluidos.



## ÍNDICE

<b>RESUMEN .....</b>	<b>5</b>
<b>INTRODUCCIÓN .....</b>	<b>13</b>
BRECHAS ENTRE CURRÍCULO, PEDAGOGÍA Y TECNOLOGÍAS DIGITALES .....	15
<b>JUSTIFICACIÓN .....</b>	<b>19</b>
<b>TESIS CENTRAL DEL ENSAYO .....</b>	<b>23</b>
<b>METODOLOGÍA.....</b>	<b>27</b>
<b>CAPÍTULO 1. LA ENSEÑANZA DE PROGRAMACIÓN COMO CAMPO EN DISPUTA .....</b>	<b>29</b>
ENTRE EL PARADIGMA TRADICIONAL Y LAS PEDAGOGÍAS EMERGENTES .....	33
EL DESFASE ENTRE EL APRENDIZAJE ALGORÍTMICO Y LOS MODELOS DE ENSEÑANZA .....	38
EL ESTUDIANTE DIGITAL ANTE UN AULA ANALÓGICA.....	41
<b>CAPÍTULO 2. EL CASO DE BUENAVENTURA: RADIOGRAFÍA DE LA DESERCIÓN Y BAJO RENDIMIENTO EN FUNDAMENTOS DE PROGRAMACIÓN.....</b>	<b>46</b>
FACTORES INDIVIDUALES, ESTRUCTURALES E INSTITUCIONALES ....	49
EL CONTEXTO SOCIOEDUCATIVO DEL PACÍFICO COLOMBIANO COMO CONDICIONANTE.....	53
<b>CAPÍTULO 3. LMS, TIC Y VIRTUALIDAD CRÍTICA EN LA EDUCACIÓN SUPERIOR .....</b>	<b>58</b>
CARACTERÍSTICAS DE MOODLE 2.5: LÍMITES, POTENCIALIDADES Y APROPIACIÓN .....	61
MOODLE COMO ENTORNO DE APRENDIZAJE SIGNIFICATIVO .....	65
<b>CAPÍTULO 4. CURSO VIRTUAL Y ESTRATEGIA PEDAGÓGICA: DIAGNÓSTICO, DISEÑO INSTRUCCIONAL Y VALIDACIÓN .....</b>	<b>68</b>
DIAGNÓSTICO: RECONOCER EL PROBLEMA EN SU COMPLEJIDAD.....	68
DISEÑO INSTRUCCIONAL: PLANIFICAR CON SENTIDO PEDAGÓGICO ..	69
VALIDACIÓN: EVALUAR LA EXPERIENCIA, NO SOLO LOS RESULTADOS .....	70

RECURSOS UTILIZADOS, SECUENCIA DIDÁCTICA Y EVALUACIÓN FORMATIVA.....	71
RESULTADOS DEL PILOTO: PARTICIPACIÓN, COMPRENSIÓN Y PERMANENCIA.....	75
<b>CAPÍTULO 5. ENSEÑAR PROGRAMACIÓN EN CONTEXTO: TECNO-PEDAGOGÍA, ENSEÑAR SIN REPRODUCIR EXCLUSIÓN .....</b>	<b>80</b>
LA AUTONOMÍA DEL ESTUDIANTE COMO META FORMATIVA.....	84
EL ROL DOCENTE RECONFIGURADO EN CLAVE DE MEDIACIÓN.....	87
<b>CAPÍTULO 6. MOODLE 2.5 DESDE BUENAVENTURA: UNA PROPUESTA REPLICABLE .....</b>	<b>92</b>
¿QUÉ FUNCIONÓ QUÉ FALTÓ, Y QUÉ PUEDE MEJORARSE?.....	92
CLAVES PARA LA ADAPTACIÓN EN OTRAS UNIVERSIDADES PÚBLICAS .....	94
<b>CAPÍTULO 7. EL ECOSISTEMA MOODLE EN 2025: DE LA VERSIÓN 2.5 AL LMS INTELIGENTE.....</b>	<b>98</b>
USABILIDAD, INTELIGENCIA ARTIFICIAL, ANALÍTICAS DE APRENDIZAJE Y GAMIFICACIÓN.....	100
COMPARACIÓN CRÍTICA CON CANVAS, BLACKBOARD Y GOOGLE CLASSROOM.....	103
¿TECNOLOGÍA SIN PEDAGOGÍA?: RIESGOS Y DILEMAS ACTUALES ...	106
<b>CAPÍTULO 8. RETOS FUTUROS PARA LA ENSEÑANZA DE PROGRAMACIÓN EN LA ERA DIGITAL .....</b>	<b>110</b>
POLÍTICAS INSTITUCIONALES, FORMACIÓN DOCENTE Y DISEÑO UNIVERSAL.....	111
EL POTENCIAL DE LAS MICROCREDENCIALES Y ESTRATEGIAS DE RETENCIÓN .....	114
DEL AULA TÉCNICA AL AULA CRÍTICA: RECONFIGURAR LA EDUCACIÓN EN INGENIERÍA .....	117
<b>CAPÍTULO 9. REPENSAR LA ENSEÑANZA DE PROGRAMACIÓN: SÍNTESIS Y PROPUESTAS .....</b>	<b>122</b>
DE LA EXCEPCIÓN A LA ESTRATEGIA: LECCIONES DESDE BUENAVENTURA.....	122
MOODLE COMO MEDIO, NO COMO FIN .....	123

FORMACIÓN CON SENTIDO TERRITORIAL, ÉTICO Y TRANSFORMADOR .....	124
PROPUESTA DE LINEAMIENTOS PARA UNA VIRTUALIZACIÓN INCLUSIVA Y CRÍTICA.....	126
<b>BIBLIOGRAFÍA .....</b>	<b>129</b>
<b>LAS AUTORAS.....</b>	<b>133</b>



MOODLE 2.5  
PARA FORTALECER  
FUNDAMENTOS DE  
PROGRAMACIÓN  
EN INGENIERÍA DE SISTEMAS



## Introducción

En la era digital, enseñar programación en la educación superior no es solo un reto técnico, sino una prueba crítica para la capacidad adaptativa de los sistemas educativos.

En el siglo XXI, el pensamiento computacional ha sido ampliamente reconocido como una competencia clave para la ciudadanía global y el desarrollo profesional en múltiples áreas del conocimiento (Wing, 2006; Shute et al., 2017), pues no solo permite resolver problemas con herramientas digitales, sino también modelar procesos complejos, abstraer información relevante y automatizar tareas mediante algoritmos.

Es por eso que, la enseñanza de la programación debe ocupar un lugar estratégico en los currículos universitarios, particularmente en carreras como Ingeniería de Sistemas, Ciencias Computacionales o Tecnologías de la Información.

No obstante, la incorporación efectiva de la programación como saber formativo sigue enfrentada a resistencias tanto estructurales como metodológicas.

A pesar del discurso sobre innovación y competencias digitales, muchas aulas universitarias continúan ancladas en modelos de enseñanza transmisivos, centrados en la explicación unidireccional y en la evaluación memorística.

Este rezago pedagógico se produce en un contexto donde los estudiantes ya interactúan

cotidianamente con dispositivos digitales, pero lo hacen desde una relación más instrumental que crítica, y con escasa alfabetización computacional profunda (Denning, 2009; Sentance et al., 2019). Así, se configura una paradoja preocupante: se enseña programación en entornos digitales, a estudiantes digitales, con herramientas digitales, pero desde paradigmas que no han cambiado sustancialmente desde el siglo pasado.

Esta contradicción se manifiesta de manera especialmente aguda en la asignatura Fundamentos de Programación, materia base en la mayoría de programas de Ingeniería de Sistemas.

Esta asignatura, concebida para cimentar el pensamiento algorítmico, la lógica computacional y la capacidad de abstracción técnica, se convierte frecuentemente en una de las mayores causas de deserción, repitencia o rezago académico.

Diversos estudios en Latinoamérica han documentado tasas de aprobación inferiores al 50 % en cursos iniciales de programación, incluso en universidades con buena infraestructura tecnológica (Rodríguez, García & Medina, 2021). Tal situación no puede explicarse únicamente por la dificultad intrínseca de los contenidos, sino por una confluencia de factores que van desde debilidades en la formación matemática previa, hasta el uso de metodologías de enseñanza poco activas y la ausencia de mediaciones tecnológicas significativas (Valencia-Hernández & Salinas, 2019).

También ocurre que, en muchas instituciones, los currículos permanecen desarticulados respecto a las transformaciones del campo computacional contemporáneo. Aún se enseñan lenguajes sin vínculo con el ecosistema tecnológico actual, y se priorizan aspectos sintácticos por encima del desarrollo de la lógica algorítmica o el pensamiento estructurado.

A esto se suma la escasa formación pedagógica de los docentes en áreas técnicas, quienes dominan los contenidos disciplinares, pero no siempre cuentan con herramientas didácticas para facilitar procesos de enseñanza-aprendizaje efectivos (Robles & Monge, 2017).

En síntesis, el desafío de enseñar programación en la universidad del siglo XXI es más profundo de lo que aparenta. No se trata simplemente de mejorar el nivel técnico de los estudiantes, sino de rediseñar las condiciones pedagógicas, curriculares y tecnológicas en las que ocurre el aprendizaje.

El lenguaje de la programación no puede seguir enseñándose como un código cifrado reservado a unos pocos iniciados. Debe ser concebido como un lenguaje de creación, comprensión y transformación del mundo digital, y como tal, requiere de enfoques que articulen tecnología, didáctica y sentido formativo.

## **Brechas entre currículo, pedagogía y tecnologías digitales**

Uno de los aspectos más críticos en la enseñanza universitaria de la programación es la distancia

persistente entre el currículo formal, las prácticas pedagógicas reales y la apropiación efectiva de las tecnologías digitales disponibles.

Aunque muchas universidades han avanzado en la adopción de plataformas de gestión del aprendizaje como Moodle, esta incorporación tecnológica ha sido, en múltiples casos, más administrativa que pedagógica. Y es que el entorno virtual ha sido frecuentemente utilizado como un simple repositorio de archivos PDF, guías de lectura o enlaces externos, sin mediación didáctica significativa ni diseño instruccional orientado al aprendizaje activo.

Este uso instrumental y lineal de las plataformas LMS limita sus posibilidades transformadoras, pues no podemos reducir un aula virtual a un espacio de almacenamiento digital, sino como un entorno pedagógico donde se dinamiza la interacción entre estudiantes, contenidos y docentes, y donde la evaluación, la retroalimentación y la construcción de sentido ocurren de manera situada y significativa. Sin embargo, en la práctica cotidiana, muchos cursos siguen operando con una lógica de “digitalización” del contenido, sin modificar los fundamentos pedagógicos que los sostienen.

Este desfase se agudiza en escenarios donde confluyen precariedad tecnológica, baja formación docente en TIC y ausencia de políticas institucionales que incentiven la innovación educativa.

En contextos como el de la Universidad del Pacífico, el uso limitado de Moodle 2.5 por parte de los docentes no se explica por falta de acceso, sino por un escaso conocimiento de sus funcionalidades pedagógicas, la falta de formación en diseño instruccional y la débil articulación entre tecnología y currículo.

Incluso el propio sistema AVAS —implementado sobre Moodle 2.5— permaneció subutilizado durante varios semestres, a pesar de estar disponible para la totalidad de la comunidad académica.

En este contexto, resulta urgente reconocer que la tecnopedagogía no puede ser concebida como un lujo reservado para instituciones con alta conectividad y presupuestos holgados. Por el contrario, en territorios con restricciones de acceso y condiciones estructurales adversas, como el litoral Pacífico colombiano, la integración significativa de tecnologías educativas se convierte en una herramienta vital para ampliar las oportunidades de aprendizaje, mejorar la calidad de la enseñanza y cerrar brechas históricas.

Pero dicha integración no ocurre de manera automática ni neutral: requiere una transformación profunda de las concepciones sobre la enseñanza, el rol del docente y la participación activa del estudiante en su propio proceso de formación.

Los desafíos no se limitan a la infraestructura o al software, sino que atraviesan el diseño curricular, la capacitación docente y las condiciones

institucionales que habilitan o inhiben el cambio educativo. En palabras de Caicedo & Vásquez, “para que plataformas como Moodle 2.5 cumplan su promesa pedagógica, deben ser concebidas como entornos de aprendizaje significativos, que permitan la interacción, la retroalimentación constante y la evaluación formativa” (Caicedo & Vásquez, 2016, p. 34).

Esto implica pues, repensar no solo el qué se enseña, sino cómo y para qué se enseña, especialmente en áreas como Fundamentos de Programación, donde la lógica computacional puede convertirse en una herramienta de empoderamiento si es mediada pedagógicamente con sentido crítico y contextual.

## Justificación

Pensar la enseñanza de la programación únicamente en términos de eficacia técnica o cobertura curricular es desconocer su dimensión social y política. En contextos como el latinoamericano, y de manera particular en regiones históricamente excluidas como el Pacífico colombiano, educar en programación no es solo formar futuros ingenieros: es abrir posibilidades de permanencia, agencia y transformación en territorios atravesados por la violencia, la exclusión y la desigualdad.

La equidad educativa, en este sentido, no se limita al acceso a la matrícula, sino que exige condiciones reales de aprendizaje, herramientas culturalmente pertinentes y estrategias pedagógicas que reconozcan la diversidad de trayectorias formativas de los estudiantes.

En la Universidad del Pacífico, el bajo rendimiento en la asignatura Fundamentos de Programación no puede atribuirse exclusivamente a las dificultades propias del lenguaje computacional.

Entre 2015 y 2016, más del 60 % de los estudiantes de primer semestre reprobó la asignatura, con niveles de aprobación alarmantemente bajos, incluso entre quienes la aprobaron con nota mínima (p. 16). Estas cifras reflejan algo más profundo que un problema disciplinar: evidencian un sistema que no ha logrado generar condiciones pedagógicas y tecnológicas adecuadas para enseñar uno de los pilares del pensamiento computacional en un contexto de inequidad estructural.

Los factores que contribuyen a este fenómeno son múltiples. Por un lado, están las debilidades formativas de base: carencias en matemáticas, comprensión lectora y habilidades TIC que los estudiantes arrastran desde niveles anteriores del sistema educativo. Por otro, aparecen las barreras materiales: falta de acceso a libros, recursos de consulta o equipos adecuados, que condicionan el ritmo y la calidad del aprendizaje; y finalmente, se encuentran las limitaciones institucionales: uso marginal de la plataforma Moodle, escasa formación del profesorado en entornos virtuales, y modelos pedagógicos centrados en la exposición magistral más que en la construcción participativa del conocimiento.

En este marco, implementar un curso virtual de Fundamentos de Programación en Moodle 2.5 no fue simplemente una estrategia de modernización, sino una respuesta concreta a un problema de exclusión educativa.

Fue una apuesta por democratizar el conocimiento a través de una mediación tecnológica apropiada, que permite a los estudiantes acceder de forma flexible a los contenidos, avanzar a su propio ritmo y fortalecer sus competencias digitales sin depender exclusivamente del aula presencial. El curso no sustituye la interacción con el docente, sino que la complementó con recursos estructurados, ejercicios interactivos, retroalimentación continua y un entorno de aprendizaje más accesible.

Pero la importancia de esta experiencia no radica solo en su efectividad local, sino en lo que revela: que

las tecnologías, cuando se articulan con sentido pedagógico y conocimiento del contexto, pueden convertirse en herramientas de equidad. El proyecto se alineó con los principios de la Ley 1341 de 2009, que promueve el acceso y uso pedagógico de las TIC en todos los niveles del sistema educativo colombiano, y se enmarca en una visión transformadora de la universidad pública como agente de inclusión social.

Por eso, este ensayo asume que el problema de la enseñanza de la programación no puede reducirse a la selección del lenguaje o a la calidad del software utilizado. Lo que está en juego es la posibilidad de construir una universidad más justa, donde aprender a programar no sea privilegio de quienes ya llegan con ventajas, sino una oportunidad real para quienes, históricamente, han sido desplazados del centro del sistema.

La equidad, en este campo, se juega en el diseño del aula, en la forma de evaluar, en el tipo de interacción que se promueve, y en la capacidad de adaptar las herramientas disponibles a las realidades concretas de los estudiantes.



## Tesis central del ensayo

La enseñanza de la programación en la educación superior enfrenta un reto que va más allá de la cobertura curricular o el dominio técnico de los lenguajes computacionales.

En contextos históricamente excluidos, como el litoral Pacífico colombiano, el acceso desigual a recursos tecnológicos, la persistencia de modelos pedagógicos transmisivos y la desarticulación entre currículo, plataforma y contexto, configuran un escenario adverso para el aprendizaje significativo del pensamiento computacional (Caicedo & Vásquez, 2016, pp. 16–18; Valencia-Hernández & Salinas, 2019).

Este ensayo sostiene que las limitaciones en la enseñanza universitaria de la programación no son, en esencia, tecnológicas, sino pedagógicas y contextuales.

Plataformas como Moodle 2.5, aunque consideradas obsoletas en comparación con LMS más recientes, pueden convertirse en entornos de aprendizaje potentes si se articulan con un diseño didáctico contextualizado, con estrategias activas de enseñanza y con un enfoque inclusivo (Bousboula, El Fadli & Khaldi, 2025, p. 2820; Dougiamas, 2022; Trivedi, 2023a).

Esta propuesta se fundamenta en el principio de que la calidad del aprendizaje no depende exclusivamente del avance tecnológico, sino de la intencionalidad pedagógica con la que se integran

las herramientas disponibles, según han sugerido autores como Alotaibi (2024), al abordar el uso ético y crítico de la inteligencia artificial en los LMS, y Schaffhauser (2023), al referirse al potencial de las analíticas de aprendizaje cuando están al servicio de una pedagogía centrada en el estudiante.

Este ensayo busca dar elementos para responder a las siguientes preguntas: ¿cómo transformar la enseñanza de la programación en escenarios de exclusión educativa? ¿Qué rol pueden cumplir las plataformas virtuales como Moodle 2.5 cuando se aplican con enfoque pedagógico riguroso? ¿Cuáles son las condiciones necesarias para que la virtualidad aporte a la equidad y no a la reproducción de desigualdades?

Para ello, analizamos críticamente los desafíos estructurales en la enseñanza universitaria de la programación, particularmente en territorios como el Pacífico colombiano, donde los niveles de aprobación en Fundamentos de Programación revelan un patrón persistente de fracaso académico (Caicedo & Vásquez, 2016, p. 16; Rodríguez, García & Medina, 2021).

Exploramos el potencial pedagógico de Moodle 2.5 como plataforma de mediación tecnopedagógica, considerando su adaptabilidad, modularidad y posibilidad de configuración, incluso en contextos de infraestructura limitada (Moodle.org, s.f.; Dougiamas, 2022).

Examinamos el caso de Buenaventura como ejemplo de innovación educativa territorializada,

destacando las decisiones de diseño instruccional, las estrategias de acompañamiento docente y los resultados preliminares en términos de participación y rendimiento.

Y, finalmente, proponemos lineamientos para el rediseño pedagógico de cursos de programación, integrando estrategias activas, herramientas accesibles y criterios de inclusión, en línea con las recomendaciones actuales sobre equidad, analíticas de aprendizaje, accesibilidad y gamificación (Alotaibi, 2024; Trivedi, 2023b; Schaffhauser, 2023).



## Metodología

El presente ensayo es el resultado de un estudio analítico y reflexivo que se inscribe en la intersección entre pedagogía, tecnología y equidad educativa. Su enfoque metodológico es de tipo cualitativo, crítico y documental, orientado a comprender las condiciones, tensiones y posibilidades que atraviesan la enseñanza universitaria de la programación en contextos de exclusión, con especial atención a regiones como el Pacífico colombiano.

Para esto, basamos la estrategia de análisis en tres ejes articuladores:

En primer lugar, en la revisión crítica de literatura especializada sobre plataformas virtuales de aprendizaje, sobre enseñanza de programación, innovación educativa, pensamiento computacional, tecnopedagogía y sobre políticas de inclusión digital.

En segundo lugar, en la lectura contrastiva y argumentativa de estudios comparativos y técnicos sobre plataformas LMS, como los realizados por Bousboula et al. (2025), Trivedi (2023a), Colman & Salas (2025) y Dougiamas (2022), que permiten ubicar el caso de Moodle 2.5, en relación con otras herramientas del ecosistema educativo digital contemporáneo.

En tercer lugar, en el análisis crítico de una experiencia educativa documentada en la Universidad del Pacífico, en el libro de Caicedo Guerrero & Vásquez Olaya (2016) *Creación de un*

*curso virtual para mejorar las clases de Fundamentos de Programación en estudiantes de primer semestre del programa de Ingeniería de Sistemas de la Universidad del Pacífico mediante la herramienta Moodle 2.5;* un documento ofrece información sobre el diseño, la implementación y evaluación de un curso virtual de Fundamentos de Programación, en un contexto de precariedad tecnológica, lo que constituye un buen punto de partida para reflexionar sobre modelos pedagógicos aplicables en condiciones similares.

La construcción del argumento de este texto se estructura pues, desde un enfoque hermenéutico-interpretativo.

En cuanto a las fuentes utilizadas, estas se analizan no como insumos cerrados, sino como materiales que dialogan entre sí para permitir una visión crítica e integrada del fenómeno educativo, y orientar de este modo el rediseño de experiencias de enseñanza de programación más inclusivas, efectivas y contextualmente pertinentes.

De manera deliberada, el ensayo evita caer en una sistematización técnica de una experiencia puntual. Su propósito no es replicar un caso, sino extraer de él preguntas, criterios y aprendizajes que permitan repensar la relación entre tecnologías digitales y formación en ingeniería en clave de justicia educativa.

# Capítulo 1. La enseñanza de programación como campo en disputa

La enseñanza universitaria de la programación representa uno de los ejes más relevantes en la formación de ingenieros y profesionales del ámbito tecnológico en el siglo XXI.

Su importancia no solo radica en el aprendizaje técnico de un lenguaje computacional, sino en el desarrollo del pensamiento algorítmico, la capacidad para abstraer problemas complejos y traducirlos en estructuras lógicas; y en la consolidación de habilidades como la resolución autónoma de problemas, la colaboración interdisciplinar y la comprensión crítica del entorno digital.

Sin embargo, este campo formativo se encuentra atravesado por profundas tensiones estructurales. Durante décadas, la programación ha sido enseñada bajo un enfoque técnico, centrado en la repetición de instrucciones y la memorización de sintaxis, muchas veces desprovisto de contexto, creatividad o reflexión pedagógica.

A pesar de que el pensamiento computacional ha sido reconocido como una competencia clave en la formación contemporánea (Wing, 2006; Shute, Sun & Asbell-Clarke, 2017), la mayoría de las universidades aún operan con modelos educativos anclados en paradigmas tradicionales que no se

corresponden con las formas actuales de aprender y enseñar.

Este desfase se acentúa en escenarios de exclusión estructural como los que enfrentan muchas instituciones públicas en regiones periféricas, donde los retos no se reducen al aula, sino que incluyen precariedades materiales, limitaciones en la formación docente y escasa infraestructura tecnológica.

Allí, la enseñanza de programación —lejos de ser una oportunidad para el desarrollo— puede transformarse en una trampa que condena a los estudiantes al fracaso, especialmente en los primeros semestres. Así lo evidencia el caso de la Universidad del Pacífico, donde más del 60 % de los estudiantes reprobó la asignatura de Fundamentos de Programación entre 2015 y 2016, lo que evidencia un problema de fondo más allá del dominio técnico del lenguaje (Caicedo & Vásquez, 2016, p. 16).

A lo largo de este capítulo se plantea que la programación, más que un contenido, es un campo de disputa entre lógicas educativas. Por un lado, persiste un enfoque transmisivo, centrado en la enseñanza lineal, la ejecución de ejercicios repetitivos y la evaluación estandarizada; y por otro, emergen propuestas pedagógicas que buscan integrar tecnologías digitales, aprendizaje activo, gamificación, analíticas de aprendizaje y enfoques constructivistas al proceso formativo.

Sin embargo, la transición entre un modelo y otro no ocurre de manera natural ni espontánea. Requiere voluntad institucional, formación docente y un rediseño estructural del modo en que se conciben los cursos, las plataformas y la relación pedagógica.

Tal tensión entre paradigmas genera tres problemáticas profundamente interconectadas y se constituyen en dimensiones críticas. La primera es la persistencia del modelo tradicional, que continúa privilegiando la exposición magistral, la instrucción secuencial y el control sintáctico del código. En este enfoque, el estudiante es un ejecutor pasivo de estructuras preestablecidas, sin espacio para el error, la creatividad o el aprendizaje autónomo.

Investigaciones recientes han mostrado que este modelo tiene un impacto limitado sobre la comprensión conceptual y la motivación del estudiantado (Valencia-Hernández & Salinas, 2019; Sentance et al., 2019).

La segunda problemática es el desfase entre las capacidades cognitivas que se espera desarrollar — como la abstracción algorítmica o la resolución estructurada de problemas— y los modelos pedagógicos que realmente se aplican.

A menudo, el currículo no ofrece un recorrido progresivo que permita al estudiante construir su pensamiento computacional, sino que lo expone prematuramente a tareas complejas, sin mediaciones didácticas adecuadas. Como lo advierten Bousboula, El Fadli y Khaldi (2025), la sola presencia de una plataforma LMS no garantiza el

aprendizaje significativo si no se acompaña de un diseño instruccional adaptado al tipo de competencias que se busca fomentar.

La tercera dimensión crítica está en la desconexión entre las prácticas docentes universitarias y las formas contemporáneas de aprendizaje digital.

Los estudiantes actuales llegan a la universidad con una relación cotidiana, aunque fragmentaria, con las tecnologías digitales. No obstante esto, las aulas universitarias continúan operando bajo una lógica analógica, muchas veces desconectada de la cultura digital de los estudiantes.

En este contexto, se refuerza un desencuentro entre el sujeto que aprende y el entorno que se le ofrece para aprender. Esta brecha se amplifica cuando no se integran recursos como retroalimentación inmediata, espacios de participación en línea o itinerarios de aprendizaje personalizados, todos ellos disponibles en versiones actualizadas de Moodle (Trivedi, 2023a; Dougiamas, 2022).

La experiencia educativa documentada por Caicedo y Vásquez (2016) en la Universidad del Pacífico muestra que incluso una versión anterior de Moodle —la 2.5— puede ser reapropiada de forma significativa, si se integra con un enfoque pedagógico situado.

El uso de recursos como foros de discusión, cuestionarios automatizados, actividades prácticas y materiales diseñados para contextos de baja conectividad permitió una mejora en la participación, el rendimiento y la percepción del

aprendizaje entre los estudiantes del curso piloto (pp. 49–52).

Pero más allá de los resultados puntuales, esta experiencia demuestra que el cambio no depende exclusivamente de la tecnología, sino del marco pedagógico que le da sentido.

En suma, este capítulo invita a pensar la enseñanza de programación no como un campo técnico neutro, sino como un espacio de disputa entre formas de concebir el conocimiento, la relación docente-estudiante y el papel de la universidad pública.

Superar el modelo tradicional no implica abandonar el rigor técnico, sino complementarlo con estrategias didácticas capaces de responder a los retos de inclusión, permanencia y justicia educativa en contextos donde aprender a programar puede marcar la diferencia entre continuar o abandonar la universidad.

## **Entre el paradigma tradicional y las pedagogías emergentes**

La enseñanza de la programación en los entornos universitarios ha estado, durante décadas, profundamente influida por un paradigma tradicional que concibe el aprendizaje como la adquisición lineal y progresiva de conocimientos técnicos.

Este enfoque ha moldeado no solo los contenidos curriculares, sino también las metodologías empleadas, las formas de evaluación y la relación entre docentes y estudiantes. En el caso de las

carreras de ingeniería, esta tradición se ha expresado en prácticas de enseñanza centradas en la explicación magistral, la repetición de ejercicios sintácticos y la resolución de problemas con soluciones cerradas y predeterminadas.

Según Caicedo y Vásquez (2016), esta lógica instruccional sigue dominando muchos cursos de Fundamentos de Programación, en los que el estudiante se convierte en un ejecutor mecánico de código más que en un constructor activo de significados (pp. 33-34).

El paradigma tradicional asume que aprender a programar consiste, principalmente, en memorizar estructuras, aplicar reglas sintácticas y dominar comandos a través de la práctica reiterada. Esta concepción privilegia la precisión formal y el resultado correcto por encima de la comprensión conceptual, la creatividad algorítmica o el proceso de razonamiento que subyace a la solución de problemas.

El rol del docente, en este modelo, es el de transmisor autorizado del conocimiento, mientras que el estudiante ocupa una posición pasiva, centrada en la recepción y reproducción del saber. La evaluación, por su parte, suele reducirse a pruebas escritas o ejercicios de codificación que validan la correcta aplicación de procedimientos, sin necesariamente fomentar la reflexión crítica o la transferencia de aprendizajes a contextos diversos.

Esta visión ha comenzado a ser cuestionada por una creciente corriente de pensamiento pedagógico que

apuesta por enfoques más activos, situados y centrados en el estudiante. Se trata de lo que diversos autores han denominado *pedagogías emergentes*, un conjunto de propuestas que, sin conformar un modelo único, comparten la convicción de que el aprendizaje debe ser significativo, colaborativo y contextualizado (Valencia-Hernández & Salinas, 2019). En el campo específico de la programación, estas pedagogías promueven el uso de metodologías como el aprendizaje basado en proyectos (ABP), la gamificación, la programación por pares, el uso de entornos gráficos y visuales, y la integración de herramientas tecnológicas como mediadores del proceso educativo.

En este marco, aprender a programar no se reduce a la apropiación de un lenguaje computacional, sino que implica el desarrollo del pensamiento computacional como una forma de organizar la experiencia, analizar problemas complejos, construir modelos abstractos y plantear soluciones de forma estructurada.

Esta noción, impulsada por autoras como Wing (2006), sitúa la programación en un lugar central del aprendizaje universitario, no solo como una competencia técnica sino como una práctica intelectual transversal y estratégica. Desde esta perspectiva, el estudiante debe ser concebido como un agente activo que experimenta, se equivoca, reflexiona y construye soluciones en interacción con otros.

Ahora bien, el tránsito entre el paradigma tradicional y las pedagogías emergentes no se produce de forma automática ni natural. Supone un proceso de transformación que implica revisar profundamente las concepciones del profesorado sobre la enseñanza, repensar los roles en el aula, rediseñar los cursos desde sus objetivos hasta su evaluación, e integrar las tecnologías digitales no como complemento, sino como parte estructural de las experiencias formativas.

Como lo muestran Caicedo y Vásquez (2016), incluso en contextos con plataformas tecnológicas limitadas —como el uso de Moodle 2.5 en la Universidad del Pacífico— es posible construir experiencias pedagógicas innovadoras si se cuenta con claridad didáctica, acompañamiento institucional y conocimiento del contexto (pp. 34–35, 49–50).

Moodle, en este sentido, ofrece una serie de funcionalidades que, si bien son muchas veces subutilizadas, pueden facilitar el desarrollo de estrategias activas de enseñanza: actividades condicionales, foros de discusión, tareas con rúbricas, cuestionarios adaptativos, retroalimentación automatizada, entre otros.

El problema no radica, entonces, en la plataforma, sino en la manera como esta es integrada —o no— al diseño pedagógico. Tal como advierten Dougiamas (2022) y Trivedi (2023a), incluso versiones anteriores de Moodle pueden generar entornos de aprendizaje significativos si se utilizan con intencionalidad y visión pedagógica.

La experiencia en Buenaventura documentada por Caicedo y Vásquez (2016) muestra que, cuando se supera el uso meramente instrumental de Moodle como “contenedor de archivos” y se lo concibe como espacio para la interacción, la retroalimentación y la autonomía, es posible impactar positivamente en el aprendizaje.

En su estudio, las autoras demuestran que un rediseño didáctico del curso de Fundamentos de Programación permitió no solo mejorar el desempeño académico de los estudiantes, sino también aumentar su motivación, participación y confianza frente a una asignatura históricamente percibida como difícil y excluyente (pp. 49–52).

En definitiva, enseñar programación desde una perspectiva transformadora requiere más que cambiar contenidos o actualizar versiones de software. Implica adoptar una visión pedagógica que coloque al estudiante en el centro del proceso, reconozca la diversidad de trayectorias y capacidades, y utilice las tecnologías como mediaciones para construir conocimiento.

En contextos de exclusión, como el litoral Pacífico, este enfoque no solo es deseable: es urgente. Porque allí, más que en ningún otro lugar, enseñar programación puede ser una vía para democratizar el conocimiento, fortalecer la permanencia universitaria y construir condiciones de justicia educativa.

## El desfase entre el aprendizaje algorítmico y los modelos de enseñanza

Enseñar programación supone mucho más que transmitir una técnica o entrenar una habilidad instrumental: implica acompañar la construcción de un tipo de pensamiento altamente estructurado, abstracto y progresivo.

El pensamiento algorítmico exige que el estudiante aprenda a modelar situaciones complejas, a descomponer problemas, identificar patrones, diseñar secuencias lógicas y evaluar resultados desde una perspectiva computacional (Wing, 2006; Shute, Sun & Asbell-Clarke, 2017).

Este tipo de aprendizaje no ocurre por exposición ni por repetición mecánica, sino mediante procesos reflexivos, interactivos y contextualizados que requieren un andamiaje didáctico intencionado.

Sin embargo, existe un desfase profundo entre esta naturaleza cognitiva del aprendizaje algorítmico y los modelos de enseñanza que predominan en muchas universidades; en particular, en las asignaturas introductorias de programación es frecuente encontrar una pedagogía centrada en la explicación lineal de conceptos, seguida por ejercicios técnicos que los estudiantes deben resolver sin una comprensión integral del problema.

La lógica que subyace a estos modelos es la de “enseñar primero la teoría y luego pasar a la práctica”, bajo el supuesto de que la programación se aprende acumulando reglas, sintaxis y

estructuras que luego podrán ser aplicadas (Caicedo & Vásquez, 2016, pp. 33–35).

Este modelo, sin embargo, desconoce que aprender a programar implica un proceso complejo de apropiación que requiere, entre otras cosas, la posibilidad de experimentar, cometer errores, recibir retroalimentación y construir progresivamente estructuras de sentido.

Como han demostrado autores como Sentance et al. (2019), los estudiantes necesitan avanzar de forma gradual desde la comprensión del problema hasta la codificación, pasando por etapas intermedias que incluyan la predicción de comportamientos, la exploración guiada y la reflexión metacognitiva. Sin estos componentes, el aprendizaje algorítmico se ve truncado, y la programación se convierte en una serie de pasos oscuros cuya lógica interna permanece opaca para el estudiante.

El desfase entre lo que se espera que los estudiantes desarrollen —pensamiento computacional, capacidad de abstracción, resolución de problemas— y lo que realmente sucede en las aulas tiene consecuencias serias.

En primer lugar, genera altos niveles de ansiedad, frustración y deserción, especialmente en contextos donde los estudiantes no cuentan con una base sólida en matemáticas o competencias digitales previas.

Y en segundo lugar, favorece la reproducción de desigualdades, ya que quienes han tenido experiencias previas en programación o acceso a

tecnologías cuentan con ventajas sustanciales frente a quienes se enfrentan por primera vez al lenguaje computacional (Rodríguez, García & Medina, 2021).

Frente a este panorama, las plataformas digitales de aprendizaje podrían ofrecer alternativas para reconfigurar la enseñanza de la programación de forma más progresiva y personalizada. Moodle, por ejemplo, permite diseñar secuencias de aprendizaje condicionales, donde los estudiantes avanzan a su propio ritmo a medida que consolidan conocimientos previos; facilita la implementación de cuestionarios automatizados con retroalimentación inmediata; y permite construir espacios de simulación y discusión colaborativa (Dougiamas, 2022; Trivedi, 2023a).

Sin embargo, como advierten Bousboula, El Fadli y Khaldi (2025), la efectividad de estas herramientas depende enteramente del diseño pedagógico que las articule. Usar Moodle como simple depósito de archivos o canal de entrega de tareas no resuelve el problema; al contrario, puede reforzar las lógicas tradicionales que ya se mostraron ineficaces.

La experiencia documentada por Caicedo y Vásquez (2016) en la Universidad del Pacífico muestra que, incluso en condiciones de limitaciones tecnológicas, es posible diseñar un curso de programación que responda mejor a las dinámicas reales del aprendizaje algorítmico. En su propuesta, se articulan ejercicios prácticos, recursos multimedia, secuencias ordenadas por niveles de dificultad y estrategias de autoevaluación que permitieron a los

estudiantes progresar en su comprensión del código sin sentirse abrumados desde el inicio (pp. 49–52).

El curso no eliminó la complejidad de aprender a programar, pero sí la hizo más accesible, comprensible y abordable para una población con trayectorias escolares diversas.

En síntesis, el desfase entre el aprendizaje algorítmico y los modelos tradicionales de enseñanza no es solo una cuestión metodológica, sino una expresión de una mirada restringida sobre el acto educativo.

Enseñar a programar debe entenderse como un proceso pedagógico integral que requiere acompañamiento, personalización, progresividad y evaluación formativa. La virtualidad no es una solución en sí misma, pero puede ser un medio potente si se diseña con sentido pedagógico, en función de las necesidades y características del estudiantado. Superar este desfase no es simplemente una mejora técnica: es una apuesta por una enseñanza más justa, pertinente y transformadora.

## **El estudiante digital ante un aula analógica**

Uno de los desafíos más notorios —y a menudo menos abordados— en la enseñanza de la programación es el creciente desajuste entre los entornos digitales que habitan los estudiantes y las estructuras analógicas de muchas aulas universitarias.

Esta desconexión, que se expresa en las formas de comunicación, organización del tiempo, acceso al conocimiento y uso de la tecnología, no es solo un asunto técnico: es un problema pedagógico de fondo que compromete la capacidad de las instituciones para generar aprendizajes significativos, especialmente en campos como la ingeniería de sistemas, donde la actualización constante es un imperativo.

La generación actual de estudiantes universitarios ha crecido inmersa en un entorno digital diverso, interactivo y fragmentado. Utilizan plataformas móviles, redes sociales, motores de búsqueda, aplicaciones multimedia y servicios en la nube con una fluidez que transforma sus modos de acceso al conocimiento.

Sin embargo, esta familiaridad tecnológica no se traduce automáticamente en habilidades computacionales profundas, y mucho menos en pensamiento algorítmico. Como señala Alotaibi (2024), existe una diferencia crucial entre ser consumidor de tecnología y ser un actor activo en la construcción de procesos tecnológicos mediante el pensamiento computacional (p. 10358).

El problema se agudiza cuando estos estudiantes ingresan a entornos universitarios que aún operan con lógicas propias de una cultura analógica: clases magistrales centradas en el docente, uso restringido de la tecnología, materiales impresos o digitalizados sin diseño instruccional, y escasa interacción significativa.

En este tipo de aulas, el estudiante digital no solo encuentra una barrera cognitiva, sino también un desajuste cultural. Las metodologías que se le ofrecen no responden a sus expectativas de dinamismo, personalización, colaboración o inmediatez en la retroalimentación (Schaffhauser, 2023).

Este desfase se hace particularmente evidente en los cursos de introducción a la programación, que en muchos casos presentan un entorno doblemente distante: por un lado, el lenguaje computacional abstracto que el estudiante no ha explorado antes, y por otro, una forma de enseñanza rígida que no se ajusta a sus formas previas de aprender.

Como lo documentan Caicedo y Vásquez (2016), en contextos como el de Buenaventura, donde las trayectorias educativas suelen estar atravesadas por brechas estructurales, esta distancia entre cultura tecnológica cotidiana y aula universitaria puede traducirse en frustración, fracaso y abandono académico (pp. 29–31).

Frente a este panorama, las plataformas de gestión del aprendizaje como Moodle pueden ofrecer puentes entre la experiencia digital del estudiante y las exigencias formativas de la universidad. Herramientas como los foros asincrónicos, las actividades interactivas, los cuestionarios con retroalimentación inmediata y los recursos multimedia integrados permiten diseñar entornos más cercanos a las formas actuales de aprender.

Sin embargo, como advierten Dougiamas (2022) y Trivedi (2023a), la clave no está en la tecnología en sí misma, sino en cómo se utiliza pedagógicamente para construir sentido, motivar la participación y personalizar el aprendizaje.

La experiencia desarrollada con Moodle 2.5 en la Universidad del Pacífico, aunque limitada en términos de funcionalidades frente a versiones más recientes, mostró que incluso en condiciones tecnológicas básicas es posible ofrecer una experiencia formativa más cercana al perfil del estudiante digital.

A través de una interfaz ordenada, contenidos accesibles, secuencias progresivas y tareas integradas con herramientas audiovisuales, el curso logró aumentar el nivel de interacción y apropiación de los contenidos, especialmente entre estudiantes que habían fracasado previamente en su aprendizaje de la programación (Caicedo & Vásquez, 2016, pp. 49–52).

En suma, cerrar la brecha entre el estudiante digital y el aula analógica requiere más que introducir dispositivos en el salón de clase. Implica reconfigurar la experiencia educativa desde la comprensión profunda de cómo aprenden hoy los jóvenes, cuáles son sus referencias cognitivas, qué esperan de un proceso formativo, y cómo puede la universidad aprovechar sus entornos virtuales para acompañar, orientar y potenciar ese aprendizaje. No hacerlo es condenar al sistema educativo a una creciente irrelevancia, y a muchos estudiantes — especialmente en contextos de vulnerabilidad— a

ser excluidos de una cultura digital que podrían habitar de forma activa y transformadora.

## Capítulo 2. El caso de Buenaventura: Radiografía de la deserción y bajo rendimiento en Fundamentos de Programación

La asignatura Fundamentos de Programación representa, en los planes curriculares de Ingeniería de Sistemas, mucho más que una materia introductoria: es la piedra angular sobre la que se construye la comprensión del pensamiento computacional, entendido como la capacidad de formular, descomponer y resolver problemas mediante modelos lógicos, estructurados y reproducibles.

En este sentido, su importancia no radica únicamente en la enseñanza de un lenguaje de codificación, sino en la formación de una manera específica de pensar, indispensable para la lógica del diseño de software, el desarrollo de sistemas y la ingeniería computacional en general (Wing, 2006; Sentance et al., 2019).

Sin embargo, esta importancia estratégica contrasta con los resultados académicos registrados en múltiples instituciones, particularmente en contextos marcados por desigualdades educativas como Buenaventura.

En la Universidad del Pacífico, por ejemplo, el curso de Fundamentos de Programación ha sido identificado como uno de los principales factores

asociados a la deserción temprana. Los datos presentados por Caicedo y Vásquez (2016) revelan que entre los años 2015 y 2016, más del 60 % de los estudiantes que cursaban la asignatura en primer semestre la reprobaban, con un impacto directo en su decisión de continuar o abandonar el programa (p. 16).

Este indicador, lejos de ser una excepción coyuntural, respondía a una tendencia estructural que se repetía semestre tras semestre, sin que existieran mecanismos institucionales eficaces para revertirla.

El fenómeno no se explicaba exclusivamente por la complejidad técnica del contenido. En realidad, el alto índice de reprobación revelaba un cúmulo de fallas sistémicas: vacíos formativos acumulados desde la educación media, falta de preparación en lógica formal y resolución de problemas, escasa familiaridad con los entornos digitales, y metodologías de enseñanza inadecuadas para las características de ingreso de la población estudiantil.

El resultado era un desfase profundo entre lo que exigía la asignatura y lo que los estudiantes estaban en condiciones reales de asumir.

Este desajuste generaba, a su vez, efectos en cadena. El bajo rendimiento en Fundamentos de Programación se traducía en experiencias de fracaso académico temprano que minaban la motivación, debilitaban la confianza en las propias capacidades

y deterioraban el vínculo entre el estudiante y su proceso formativo.

En muchos casos, el curso dejaba de ser percibido como una oportunidad para desarrollar habilidades y pasaba a entenderse como una instancia punitiva que sancionaba el desconocimiento, sin ofrecer alternativas reales de aprendizaje.

Esta percepción alimentaba la idea de que se trataba de una “materia de filtro”, en la que no se medían capacidades reales, sino familiaridades previas con un código pedagógico al que la mayoría no había sido expuesta.

Desde esta perspectiva, el bajo rendimiento en Fundamentos de Programación no podía seguir atribuyéndose a una supuesta falta de esfuerzo individual. Se trataba, más bien, de una expresión concentrada de exclusiones acumuladas —sociales, tecnológicas, educativas— que encontraban en este curso un punto de condensación.

En ese sentido, la deserción no era el resultado de un fracaso aislado, sino la consecuencia previsible de una estructura institucional que no había sido diseñada para acoger, acompañar ni potenciar trayectorias formativas complejas.

El caso de Buenaventura, lejos de ser excepcional, funciona como espejo de una situación que atraviesa muchas universidades públicas de regiones periféricas, donde el desfase entre las exigencias curriculares y las condiciones reales de los estudiantes suele resolverse mediante el silencioso mecanismo del abandono.

Esta “normalización del fracaso” es quizás uno de los síntomas más preocupantes del modelo educativo vigente, que convierte la deserción no en un evento alarmante, sino en una estadística previsible.

La experiencia documentada en Caicedo y Vásquez (2016) en la Universidad del Pacífico buscó intervenir en esta situación mediante una propuesta pedagógica mediada por Moodle 2.5. La hipótesis era simple pero poderosa: cambiar la forma de enseñar programación podía tener efectos concretos en la permanencia estudiantil.

A partir de un rediseño instruccional que incluyó materiales adaptados, actividades progresivas, evaluación formativa y acompañamiento continuo, lograron reducir las tasas de reprobación y mejorar la percepción del curso entre los estudiantes.

No se trató de una solución definitiva, pero sí de una evidencia de que el problema no estaba en la dificultad intrínseca del contenido, sino en la forma como este era enseñado y mediado tecnológicamente.

## **Factores individuales, estructurales e institucionales**

El bajo rendimiento y la alta deserción observados en la asignatura Fundamentos de Programación en la Universidad del Pacífico no pueden comprenderse cabalmente si se analizan desde una perspectiva simplista que individualiza el fracaso.

Estos fenómenos responden a una constelación compleja de factores que interactúan entre sí, y cuya

naturaleza atraviesa dimensiones personales, sociales e institucionales. Entender estas causas en su pluralidad no solo permite dimensionar el problema, sino también diseñar intervenciones pedagógicas más integrales y situadas.

En el plano individual, el perfil de ingreso del estudiantado revela trayectorias educativas marcadas por discontinuidades, vacíos formativos y escaso acceso a experiencias previas en lógica, matemáticas y tecnologías de la información.

Como se muestra en Caicedo y Vásquez (2016), muchos de los estudiantes matriculados en primer semestre provienen de contextos escolares donde la formación básica en pensamiento lógico es débil o inexistente, y donde el acceso a computadores o internet ha sido limitado o nulo (pp. 16-17).

Esta situación genera una brecha inicial significativa frente a los requerimientos de una asignatura que demanda competencias específicas desde el inicio. Pero más aún, implica que el proceso de aprendizaje no parte de cero, sino de trayectorias educativas marcadas por la exclusión.

A esta dimensión cognitiva se suma una afectiva y motivacional. El sentimiento de extrañamiento, de “no estar a la altura”, atraviesa a muchos estudiantes en sus primeros encuentros con la programación.

La falta de comprensión de los conceptos básicos, sumada a una enseñanza rígida y poco contextualizada, genera ansiedad académica, pérdida de confianza en las propias capacidades y,

en muchos casos, abandono del esfuerzo antes de lograr una comprensión básica.

Estas reacciones no son muestras de falta de disciplina, sino el reflejo de procesos de aprendizaje frustrados que, en lugar de activar el deseo de saber, refuerzan la percepción de inadecuación.

Pero los factores individuales no explican por sí solos el problema. En realidad, se inscriben dentro de una estructura social y territorial que condiciona profundamente las posibilidades de aprender.

Buenaventura, como epicentro del Pacífico colombiano, es una ciudad históricamente excluida, con altos niveles de pobreza, precariedad en servicios públicos, violencia estructural y débil presencia del Estado. Estas condiciones se reflejan en la vida cotidiana del estudiantado: estudiantes que trabajan para sostener sus estudios, hogares sin conexión a internet, entornos familiares con limitadas posibilidades de apoyo académico.

Como lo advierten las autoras del estudio referido, estas condiciones no son anecdóticas, sino estructurales, y su desconocimiento constituye una forma de violencia institucional (Caicedo & Vásquez, 2016, pp. 25-26).

Desde esta perspectiva, los factores estructurales no se limitan al acceso material, sino que incluyen las representaciones sociales sobre quién puede y quién no puede aprender programación.

En Buenaventura, las oportunidades educativas han sido sistemáticamente negadas, y muchos

estudiantes no se reconocen como legítimos interlocutores del conocimiento computacional. Esto configura una especie de frontera simbólica que refuerza la exclusión, aún, cuando existan las condiciones objetivas para superarla.

Por último, el componente institucional también desempeña un papel decisivo. La forma en que se estructuran los cursos, la capacitación del cuerpo docente, las políticas de acompañamiento estudiantil y el uso de las tecnologías de aprendizaje determinan en buena medida las trayectorias educativas. En el caso analizado, las prácticas docentes anteriores a la intervención virtual estaban marcadas por la enseñanza expositiva, la evaluación memorística y el uso mínimo de recursos tecnológicos. Moodle, cuando era utilizado, funcionaba únicamente como repositorio de materiales, sin que existiera un diseño instruccional que articulase sus posibilidades con las necesidades reales del aprendizaje (Caicedo & Vásquez, 2016, pp. 34–35).

El rediseño pedagógico del curso, propuesto por las autoras, mostró que estas condiciones pueden ser transformadas si se entiende que la tecnología no es un fin en sí mismo, sino una mediación que, cuando se emplea con sentido pedagógico, permite construir entornos más inclusivos, flexibles y estimulantes.

La incorporación de tareas secuenciadas, retroalimentación inmediata, recursos multimedia y foros de discusión no solo modificó la experiencia de los estudiantes, sino que generó una nueva cultura

de aprendizaje, más acorde con sus realidades y potencialidades.

En síntesis, el rendimiento en Fundamentos de Programación no puede analizarse aislando al estudiante del contexto, ni al contexto de la institución. Solo una mirada que articule los factores individuales, estructurales e institucionales permitirá comprender el fenómeno de forma integral. Y, sobre todo, solo esta mirada puede orientar intervenciones capaces de interrumpir el ciclo de exclusión que ha marcado por años la enseñanza de la programación en regiones como el Pacífico colombiano

### **El contexto socioeducativo del Pacífico colombiano como condicionante**

Cualquier análisis riguroso sobre el rendimiento académico y la permanencia estudiantil en programas de ingeniería en el Pacífico colombiano debe partir de un reconocimiento explícito del territorio como factor determinante.

Lejos de ser una simple localización geográfica, Buenaventura constituye un espacio histórico y social profundamente marcado por dinámicas de exclusión estructural, que impactan de manera directa en las trayectorias educativas del estudiantado universitario. En este sentido, los problemas de deserción y bajo rendimiento no pueden analizarse como fenómenos aislados, sino como expresiones localizadas de procesos sociales más amplios.

El Pacífico colombiano, y en particular Buenaventura, ha sido históricamente una región relegada en las políticas de desarrollo nacional. Pese a su importancia estratégica como puerto y nodo económico, enfrenta déficits persistentes en indicadores clave de salud, educación, empleo y acceso a servicios públicos.

Esta marginalidad estructural tiene implicaciones directas sobre el sistema educativo: escuelas con infraestructura deficiente, planteles con carencia de docentes capacitados en áreas clave como matemáticas y tecnología, baja cobertura de internet, y trayectorias escolares marcadas por la fragmentación y la inestabilidad.

La Universidad del Pacífico, como institución pública regional, concentra en su interior estas tensiones. La mayoría de sus estudiantes provienen de contextos urbanos populares o zonas rurales cercanas, y muchos son primera generación en acceder a la educación superior.

Como se subraya en Caicedo y Vásquez (2016), esta población ingresa a la universidad con una carga simbólica y material considerable: expectativas familiares altas, pero sin redes de apoyo académico estables; disposición para aprender, pero sin hábitos de estudio consolidados; curiosidad por la tecnología, pero sin experiencia formativa en programación o lógica computacional (pp. 16–18, 25–26).

El contexto de Buenaventura también implica un marco cultural y lingüístico que no siempre es

reconocido ni valorado por el currículo académico tradicional. Los modelos pedagógicos estandarizados —frecuentemente diseñados en función de condiciones ideales y supuestos homogéneos— no contemplan la diversidad de saberes, ritmos, formas de comunicación ni referencias culturales que componen el horizonte del estudiante del Pacífico.

Esto genera un desfase entre el aula y el entorno, una especie de pedagogía desanclada del territorio que profundiza el sentimiento de extrañamiento y de no pertenencia.

En esta lógica, el rendimiento académico en asignaturas como Fundamentos de Programación no puede evaluarse sin atender a las condiciones socioeducativas que configuran el proceso formativo. Enseñar a programar en Buenaventura no es lo mismo que hacerlo en Bogotá o Medellín. Aquí, los desafíos no radican solo en la complejidad técnica del lenguaje computacional, sino en la necesidad de construir condiciones pedagógicas que reconozcan el lugar desde donde se enseña y se aprende.

Frente a estas condiciones, el uso de entornos virtuales de aprendizaje como Moodle no constituye *una panacea*, pero sí una oportunidad.

Según la experiencia documentada, incluso con una versión limitada como Moodle 2.5, es posible diseñar estrategias pedagógicas que respondan a las realidades del contexto.

La posibilidad de acceder a los contenidos fuera del aula, la secuenciación de actividades, la retroalimentación inmediata, y los foros como espacios de interacción permitieron flexibilizar el proceso formativo y adaptarlo a las condiciones materiales y culturales del estudiantado. No se trató de una simple digitalización del contenido, sino de una mediación pedagógica orientada a la equidad.

En síntesis, el contexto socioeducativo del Pacífico colombiano no debe ser considerado un obstáculo, sino un llamado urgente a la innovación pedagógica con pertinencia territorial. Superar las brechas en el aprendizaje de la programación exige reconocer que no todos los estudiantes parten del mismo lugar, ni tienen las mismas condiciones para avanzar. Moodle 2.5, pese a sus limitaciones, demostró que es posible construir propuestas inclusivas si se pone en el centro no solo el contenido, sino también al sujeto que aprende, su historia, su comunidad y su territorio.



## Capítulo 3. LMS, TIC y virtualidad crítica en la educación superior

En el actual ecosistema educativo, los entornos virtuales de aprendizaje no son ya un accesorio pedagógico, sino una infraestructura estructural sobre la cual, se diseñan, se implementan, y se evalúan los procesos formativos en prácticamente todos los niveles de la educación superior.

Las Tecnologías de la Información y la Comunicación (TIC), lejos de operar como simples medios técnicos, se han convertido en marcos que configuran no solo el acceso al conocimiento, sino también las formas de relación entre estudiantes, docentes e instituciones.

Esta transformación ha sido especialmente acelerada a raíz de la pandemia por COVID-19, pero sus raíces se extienden desde mucho antes, en una progresiva digitalización de los entornos universitarios.

Sin embargo, esta adopción acelerada de tecnologías ha sido, en muchos casos, más cuantitativa que cualitativa. Es decir, las plataformas han sido instaladas, pero no necesariamente apropiadas pedagógicamente.

Mientras algunas universidades han integrado los LMS (Learning Management Systems) en marcos de rediseño curricular e innovación didáctica, otras han replicado esquemas tradicionales de enseñanza

dentro del entorno virtual, limitando el uso de las plataformas a funciones logísticas como la distribución de archivos, el envío de anuncios o el registro de notas.

Este fenómeno ha sido ampliamente documentado en Caicedo y Vásquez (2016), quienes señalan que en la Universidad del Pacífico, antes de su intervención, Moodle 2.5 era utilizado esencialmente como una “carpeta digital” que replicaba la enseñanza presencial sin transformarla (pp. 34–35).

En este contexto, los LMS han adquirido una centralidad paradójica. Se les reconoce como herramientas clave para la gestión educativa, pero muchas veces sin que esto implique una comprensión crítica de su potencial ni de sus límites.

Moodle, por ejemplo, ha logrado consolidarse como una de las plataformas más implementadas en el mundo, especialmente en universidades públicas de América Latina, gracias a su carácter de software libre, su escalabilidad y su capacidad de personalización.

Pero esta ubicuidad no garantiza un uso pedagógicamente innovador. Como advierten Dougiamas (2022) y Trivedi (2023a), la presencia de un LMS no equivale a una mejora automática en los aprendizajes: lo decisivo es el modo en que se diseña, gestiona y vive la experiencia educativa dentro de dicho entorno.

De ahí que resulte necesario transitar de una visión tecnocrática de la virtualidad a una concepción

pedagógica y crítica. Lo que aquí denominamos *virtualidad crítica* no se refiere solo a la incorporación de tecnologías al aula, sino a la posibilidad de repensar la educación desde los entornos digitales.

Esta mirada implica, primero, comprender que la tecnología educativa nunca es neutral: toda plataforma, interfaz o funcionalidad traduce una determinada visión del aprendizaje, del rol del estudiante y del papel de la institución. Moodle, por ejemplo, permite múltiples formas de interacción, pero el modo en que estas se activan depende de las decisiones didácticas del docente y del marco institucional que lo acompaña.

En segundo lugar, la virtualidad crítica invita a preguntarse qué tipo de experiencias formativas se están diseñando. ¿Estamos formando usuarios pasivos de plataformas o agentes activos capaces de construir conocimiento colaborativamente? ¿Estamos promoviendo la memorización de contenidos o el desarrollo de habilidades de pensamiento computacional, resolución de problemas, y reflexión ética sobre el uso de la tecnología?

Estas preguntas son especialmente pertinentes en el ámbito de la enseñanza de la programación, donde el riesgo de instrumentalización es alto, y donde los aprendizajes pueden volverse meramente técnicos si no se sitúan en un marco pedagógico que priorice la comprensión y la creación.

Tercero, y de manera crucial, la virtualidad crítica debe considerar el contexto. En regiones como el Pacífico colombiano, donde persisten desigualdades estructurales en el acceso a internet, a dispositivos y a formación tecnológica básica, no se puede asumir que todos los estudiantes parten del mismo punto.

La experiencia de Caicedo y Vásquez (2016) en la Universidad del Pacífico evidencia que el diseño de cursos virtuales debe estar alineado con las condiciones reales del estudiantado: conexión intermitente, baja familiaridad con plataformas digitales, y una cultura académica aún en proceso de consolidación. En estos escenarios, la tecnología debe ser una mediación para la inclusión, no una nueva barrera que profundice las desigualdades.

En suma, el paso de los LMS como instrumentos técnicos a dispositivos pedagógicos implica un giro conceptual: no basta con disponer de la herramienta, es necesario dotarla de intencionalidad formativa. Moodle puede ser un entorno potente para la enseñanza de programación, pero solo si es apropiado con criterios pedagógicos, sensibilidad territorial y una visión crítica de la virtualidad. El desafío, entonces, no es solo tecnológico: es epistemológico, político y pedagógico.

### **Características de Moodle 2.5: límites, potencialidades y apropiación**

Moodle 2.5, versión en la que se basa la experiencia analizada en la Universidad del Pacífico, constituye un punto de inflexión en el desarrollo de esta

plataforma educativa. A pesar de haber sido lanzada en 2013, y de haber sido superada por versiones más robustas, esta edición ya contaba con funcionalidades que la hacían potencialmente útil para transformar los procesos de enseñanza y aprendizaje.

El problema, sin embargo, no radicaba tanto en sus capacidades técnicas como en el grado de apropiación pedagógica que lograban los actores institucionales.

Desde el punto de vista funcional, Moodle 2.5 ofrecía una estructura modular y jerárquica que permitía a los docentes organizar sus cursos por unidades, semanas o temas. Incorporaba herramientas de evaluación como cuestionarios automatizados, tareas individuales o grupales, foros de discusión asincrónica, wikis, glosarios colaborativos y encuestas.

También posibilitaba configurar restricciones de acceso, condicionar la visibilidad de los recursos según el progreso del estudiante y establecer criterios de seguimiento del aprendizaje. Todas estas funciones, articuladas adecuadamente, podían generar secuencias didácticas diversificadas y adaptadas a distintos estilos de aprendizaje (Caicedo & Vásquez, 2016, pp. 34–35).

Pese a estas potencialidades, la plataforma era utilizada, en la mayoría de los cursos revisados antes de la intervención pedagógica, como un simple repositorio de archivos. La interacción docente-estudiante se mantenía al margen de la virtualidad,

los ejercicios eran evaluados en papel, y la plataforma se usaba únicamente para alojar PDFs o enlaces a material externo.

Esta subutilización no respondía a una limitación de la herramienta, sino a una concepción restringida del rol de la tecnología en el aula. Como señalan Dougiamas (2022) y Trivedi (2023a), el potencial de Moodle —como de cualquier LMS— depende menos de su sofisticación técnica y más de su integración significativa al diseño pedagógico.

La experiencia documentada en la Universidad del Pacífico demostró que, incluso en un contexto de precariedad tecnológica, Moodle 2.5 podía ser reapropiado como un dispositivo didáctico potente. Para ello, fue necesario un rediseño integral del curso de Fundamentos de Programación, que incluyó: la estructuración del contenido en niveles progresivos de dificultad; el uso de foros como espacios para compartir dudas, soluciones y reflexiones; la implementación de cuestionarios automáticos con retroalimentación inmediata; y la incorporación de materiales multimediales — videos, diagramas, ejemplos interactivos— que facilitaban la comprensión de los conceptos clave.

Uno de los elementos más destacados fue la posibilidad de personalizar la experiencia de aprendizaje. La utilización de rutas condicionales, en las que el estudiante solo podía avanzar al siguiente módulo si cumplía ciertos criterios mínimos en el anterior, permitió una secuenciación más efectiva del contenido.

Además, los foros se convirtieron en una herramienta pedagógica no solo para resolver dudas, sino para fomentar la colaboración horizontal y el reconocimiento entre pares. La evaluación dejó de estar centrada en una única nota sumativa y pasó a ser continua, formativa y mediada por la retroalimentación.

Sin embargo, la apropiación plena de Moodle 2.5 no estuvo exenta de dificultades. La falta de conectividad estable, la limitada capacitación docente, y la escasa cultura institucional de innovación pedagógica fueron obstáculos que exigieron una estrategia de acompañamiento técnico y formativo sostenido.

Este proceso puso en evidencia sin embargo, que el uso pedagógico de la tecnología, y la efectividad de un LMS, depende de una triple articulación: diseño didáctico riguroso, conocimiento técnico básico por parte del profesorado y políticas institucionales que respalden el uso significativo de estas herramientas.

Para finalizar este apartado decimos que, Moodle 2.5 posee límites técnicos evidentes —una interfaz menos intuitiva y menor compatibilidad móvil frente a versiones posteriores—, pero alberga un conjunto de potencialidades que, cuando son apropiadas desde una visión pedagógica crítica, permiten diseñar entornos virtuales de aprendizaje ricos, interactivos y significativos. El éxito de esta herramienta no está dado tanto por la tecnología que ofrece, sino por el sentido pedagógico que los docentes son capaces de construir con ella.

## **Moodle como entorno de aprendizaje significativo**

La posibilidad de convertir a Moodle en un entorno de aprendizaje significativo depende del enfoque pedagógico con el cual se diseña y se implementa. Un entorno de aprendizaje significativo no se define por la cantidad de herramientas digitales utilizadas, sino por su capacidad de promover la construcción activa del conocimiento, de favorecer la reflexión crítica del estudiante y de posibilitar aprendizajes contextualizados, relevantes y sostenibles.

En este sentido, Moodle —incluso en su versión 2.5— puede desempeñar un papel central en la transformación educativa, siempre que se lo comprenda como algo más que un repositorio y se le otorgue un rol articulador dentro de una propuesta didáctica intencionada.

En la experiencia que referimos de la Universidad del Pacífico, vemos cómo es posible resignificar Moodle a partir de decisiones pedagógicas estratégicas. La estructuración del curso en unidades temáticas progresivas, el uso de retroalimentación inmediata, la incorporación de materiales visuales y audiovisuales y la creación de foros de discusión no solo reorganizaron el contenido, sino que reconfiguraron la experiencia formativa. Moodle dejó de ser una plataforma auxiliar para convertirse en el espacio principal donde ocurría el aprendizaje.

El aula virtual fue diseñada como un itinerario formativo que combinaba teoría, práctica, reflexión

y evaluación, en una lógica coherente con los ritmos y necesidades del estudiantado (pp. 45–52). Una de las claves de esta transformación fue el uso de la retroalimentación formativa.

En lugar de limitarse a corregir errores, se propuso acompañar los procesos de comprensión del estudiante mediante devoluciones escritas, explicaciones complementarias y ejercicios de recuperación. Esta estrategia, aplicada a través de los foros, los cuestionarios automatizados y los comentarios en las tareas, permitió que el estudiante no solo identificara sus equivocaciones, sino que comprendiera su lógica y tuviera herramientas para superarlas. El entorno virtual, así, adquirió una dimensión dialógica que, aunque asincrónica, multiplicó las oportunidades de aprendizaje.

Otro elemento fundamental fue la inclusión de actividades interactivas, pensadas no como juegos superficiales sino como ejercicios de aplicación contextualizada. Por ejemplo, se diseñaron problemas de programación inspirados en situaciones reales del entorno del estudiante, lo que favoreció la comprensión del sentido práctico de los algoritmos. Esta estrategia, sumada a la posibilidad de acceder a materiales en distintos formatos (textos, videos, esquemas), atendió a la diversidad de estilos de aprendizaje y favoreció una apropiación más autónoma y profunda del conocimiento.

La significatividad del entorno también se expresó en la percepción de los estudiantes. Según los

resultados recogidos por las autoras, muchos de los jóvenes que habían reprobado anteriormente la asignatura expresaron sentirse más motivados, menos ansiosos y más capaces de enfrentar los desafíos del curso gracias a la estructura clara, el acompañamiento permanente y la posibilidad de avanzar a su propio ritmo. Moodle se convirtió, en sus palabras, en “un profesor que siempre está ahí”, que guía, responde y acompaña incluso fuera del horario de clases (Caicedo & Vásquez, 2016, pp. 51-52).

Sin embargo, esta transformación no se produjo por sí sola. Requirió un cambio en la forma de concebir la enseñanza: de la transmisión al diseño, de la evaluación sumativa a la formativa, de la lógica unidireccional a la interacción. Moodle ofreció la infraestructura, pero fueron las decisiones pedagógicas las que convirtieron esa infraestructura en un entorno significativo. Esta experiencia confirma que ningún LMS es pedagógicamente potente por sí mismo: lo es en la medida en que se integra a un marco didáctico situado, reflexivo y comprometido con el aprendizaje.

Para territorios como el Pacífico colombiano, esta lección es fundamental. Moodle, incluso en sus versiones más básicas, puede contribuir a cerrar brechas si se emplea con una lógica de inclusión, pertinencia y diálogo con las condiciones locales. Diseñar entornos significativos no es una tarea técnica: es una práctica pedagógica que exige conocer al estudiante, reconocer su contexto, valorar sus saberes previos y construir puentes reales entre lo que se enseña y lo que se vive.

## Capítulo 4. Curso virtual y estrategia pedagógica: diagnóstico, diseño instruccional y validación

Transformar la enseñanza de Fundamentos de Programación en la Universidad del Pacífico no fue solo una respuesta técnica a un problema académico puntual. Fue una apuesta por replantear desde sus cimientos una experiencia de formación crítica, situada y pedagógicamente significativa.

El proceso se diseñó en fases estructuradas que integraron el diagnóstico contextual, la planificación didáctica, y la validación empírica, bajo un principio rector: la enseñanza de la programación no debe buscar solo la enseñanza de códigos, sino también propiciar condiciones para que el pensamiento computacional pueda emerger incluso en contextos de exclusión estructural.

### Diagnóstico: reconocer el problema en su complejidad

El punto de partida fue un diagnóstico detallado del curso Fundamentos de Programación, sustentado en la evidencia empírica y en la escucha activa de estudiantes y docentes. Caicedo y Vásquez (2016) identificaron que más del 60 % de los estudiantes reprobaban Fundamentos de Programación en su primer intento, y que esta asignatura operaba como un obstáculo persistente para la permanencia estudiantil (p. 16).

Sin embargo, el problema no se reducía a una cuestión técnica: entrevistas, observaciones y análisis de desempeño revelaron que los estudiantes carecían no solo de conocimientos previos en lógica y algoritmia, sino también de estrategias metacognitivas, acompañamiento personalizado y recursos digitales adecuados.

El diagnóstico no se limitó a una revisión interna del curso, sino que incorporó una lectura crítica del entorno: trayectorias escolares fragmentadas, brechas digitales, desconfianza hacia las tecnologías educativas y una cultura académica centrada en la evaluación sumativa. Esta caracterización permitió comprender que el bajo rendimiento no era un síntoma de incapacidad, sino una expresión de una pedagogía descontextualizada y de modelos de enseñanza que no dialogaban con las realidades del Pacífico colombiano.

### **Diseño instruccional: planificar con sentido pedagógico**

La segunda fase consistió en el diseño instruccional del curso virtual mediado por Moodle 2.5. A partir de los hallazgos del diagnóstico, se optó por un enfoque pedagógico progresivo, activo y formativo, estructurado en módulos temáticos de creciente complejidad. Cada módulo contenía objetivos de aprendizaje claros, materiales didácticos en distintos formatos (textuales, visuales, audiovisuales), tareas prácticas con retroalimentación, y espacios de interacción como foros colaborativos y glosarios construidos colectivamente.

El diseño instruccional no se limitó a trasladar los contenidos al entorno digital: implicó redefinir la lógica misma del curso. Se abandonó el modelo basado en clases expositivas y evaluaciones finales por una estructura basada en actividades semanales, que promovían la exploración, la experimentación, la reflexión y la aplicación contextualizada del conocimiento.

El lenguaje de programación fue introducido no como una sintaxis a memorizar, sino como una herramienta para resolver problemas cercanos a la experiencia del estudiante.

Como parte del diseño, se incorporaron principios de accesibilidad tecnológica, asegurando que todos los recursos pudieran ser descargados y reutilizados en contextos de conectividad limitada.

Asimismo, se establecieron criterios de seguimiento personalizado para identificar estudiantes en riesgo de deserción temprana y ofrecer acompañamiento oportuno desde el mismo entorno Moodle.

### **Validación: evaluar la experiencia, no solo los resultados**

La tercera fase del proceso fue la validación del curso mediante la aplicación piloto en un grupo de estudiantes de primer semestre. Esta validación fue concebida no como un ensayo técnico, sino como una instancia de retroalimentación colectiva.

Se evaluaron no solo los resultados académicos, sino también la participación, la comprensión progresiva, la percepción del curso y el impacto en

la confianza de los estudiantes respecto a su capacidad de aprender a programar.

La validación incluyó indicadores cuantitativos (tasas de aprobación, permanencia, actividad en la plataforma) y cualitativos (narrativas de los estudiantes, análisis de foros, encuestas de satisfacción).

Los resultados fueron reveladores: se observó una mejora significativa en los niveles de participación activa, un incremento en la cantidad de entregas a tiempo, y una reducción en los índices de reprobación respecto a cohortes anteriores (Caicedo & Vásquez, 2016, pp. 50-52).

Pero quizá lo más relevante fue el cambio en la percepción del curso. Los estudiantes no solo comprendieron mejor los conceptos, sino que reportaron sentirse más acompañados, menos frustrados, y más capaces de enfrentar los retos que implica aprender programación.

Este efecto subjetivo, difícil de cuantificar, fue un indicador crucial del éxito de la estrategia: demostrar que es posible enseñar de otra manera, incluso en contextos donde históricamente se ha enseñado desde la carencia.

## **Recursos utilizados, secuencia didáctica y evaluación formativa**

Como dijimos atrás, la eficacia de un entorno virtual de aprendizaje no radica únicamente en la plataforma utilizada, sino en el modo en que los recursos didácticos, la secuencia pedagógica y la

evaluación se articulan para promover un aprendizaje significativo. En el rediseño del curso Fundamentos de Programación de la Universidad del Pacífico, desarrollado con Moodle 2.5, estos tres elementos fueron repensados de manera integral, no solo para adaptar la enseñanza a la virtualidad, sino para hacerla más inclusiva, accesible y coherente con el contexto territorial y formativo del estudiantado.

En cuanto a los recursos utilizados, la propuesta se alejó de la tradicional dependencia en documentos PDF o presentaciones planas, para avanzar hacia una estrategia de diversificación mediática. Se produjeron videos tutoriales breves con explicaciones conceptuales, ejemplos y simulaciones de código, pensados para facilitar la comprensión a estudiantes con escasa experiencia en programación.

Estos materiales audiovisuales fueron acompañados por esquemas visuales, infografías, y ejercicios interactivos, diseñados con base en los principios del aprendizaje multimodal, los cuales, como afirman Mayer y Moreno (2003), potencian la retención y transferencia del conocimiento cuando se combinan estímulos auditivos y visuales de forma coordinada.

Además, se habilitaron glosarios colaborativos y espacios de trabajo asincrónicos para que los estudiantes pudieran construir definiciones propias, compartir dudas y recuperar términos clave del lenguaje computacional.

Una preocupación clave fue la accesibilidad. Todos los recursos fueron publicados en formatos livianos, descargables y compatibles con dispositivos de gama baja, una decisión pedagógica y política que reconocía las condiciones materiales del estudiantado.

Esta estrategia permitió que el curso no solo fuera técnicamente funcional, sino pedagógicamente viable en un territorio donde la conectividad es irregular y costosa. En línea con lo que proponen Burbules y Callister (2000), la mediación tecnológica no puede desligarse de las condiciones sociales en las que se implementa, y su éxito está estrechamente ligado a su pertinencia cultural y material.

En términos de secuencia didáctica, el curso fue estructurado en módulos temáticos que respondían a un principio de progresividad cognitiva. Cada unidad estaba diseñada para desarrollar habilidades específicas en un orden lógico y acumulativo, pasando de nociones básicas como tipos de datos y operadores, a estructuras condicionales, ciclos y funciones.

Esta estructura secuencial no solo respondía a una lógica curricular, sino a la necesidad de evitar el salto abrupto entre niveles de complejidad, uno de los factores que históricamente contribuía a la frustración estudiantil. Así lo señalan Sentance et al. (2019), una progresión bien diseñada en la enseñanza de programación puede reducir la ansiedad cognitiva y aumentar la confianza del

estudiante al enfrentarse a desafíos cada vez más sofisticados.

Dentro de cada módulo se incluían lecturas, videos, actividades prácticas de codificación y espacios de discusión. Estos componentes estaban alineados entre sí, lo que permitía al estudiante recorrer un camino formativo coherente, acumulativo y acompañado. Moodle 2.5, a pesar de sus limitaciones técnicas respecto a versiones posteriores, permitió organizar esta arquitectura pedagógica mediante herramientas como condicionales de acceso, seguimiento de actividades y entrega de tareas programadas.

Esta organización contribuyó a que el estudiante tuviera claridad sobre qué se esperaba en cada etapa, cómo se le evaluaría y qué recursos tenía a su disposición para avanzar.

El tercer componente central del rediseño fue la evaluación formativa. Tradicionalmente, Fundamentos de Programación había estado dominado por prácticas evaluativas sumativas, centradas en exámenes únicos de alta exigencia que no permitían el error ni el aprendizaje a partir de él.

El curso virtual propuso una ruptura con esa lógica: se construyó una evaluación orientada al acompañamiento, a la retroalimentación oportuna y al aprendizaje como proceso. Para ello, se utilizaron cuestionarios automatizados con retroalimentación inmediata, tareas revisadas con rúbricas explícitas, comentarios individualizados del docente y seguimiento del progreso en los foros.

Dicha modalidad evaluativa se basó en los principios de Black y Wiliam (2009), quienes argumentan que la evaluación formativa mejora no solo los resultados de aprendizaje, sino también la autoestima académica y la motivación del estudiante.

Los estudiantes percibieron este cambio con claridad. Como lo reportan Caicedo y Vásquez (2016), la nueva forma de evaluar fue valorada positivamente, ya que ofrecía oportunidades de mejora, reconocía los esfuerzos y permitía avanzar sin el temor paralizante al fracaso inmediato (pp. 51-52). El cambio en el modelo evaluativo fue, de hecho, uno de los aspectos más transformadores del rediseño: no solo mejoró los resultados académicos, sino que modificó la relación subjetiva del estudiante con el conocimiento, con el error y con su propia capacidad para aprender programación.

En conjunto, los recursos, la secuencia didáctica y la evaluación fueron partes de una misma estrategia: construir un entorno virtual que hiciera posible lo que antes parecía inalcanzable. Moodle 2.5 fue el soporte, pero el cambio real ocurrió en el enfoque pedagógico que dio vida a esa plataforma.

### **Resultados del piloto: participación, comprensión y permanencia**

La validación del curso Fundamentos de Programación en su versión rediseñada e implementada en Moodle 2.5 constituyó una fase final del proyecto, y una instancia crucial de evaluación del impacto pedagógico de la propuesta.

El piloto fue aplicado durante un semestre académico (Segundo semestre de 2015) en un grupo de estudiantes de primer semestre del programa de Ingeniería de Sistemas en la Universidad del Pacífico, permitiendo observar con detalle cómo respondían los estudiantes a una experiencia formativa construida desde la virtualidad, la progresividad y el acompañamiento pedagógico constante.

Uno de los primeros efectos visibles fue el aumento significativo en los niveles de participación. Mientras que en las versiones anteriores del curso una parte considerable de los estudiantes desertaba tempranamente o dejaba tareas sin entregar, en el piloto se registró una participación constante, con niveles de cumplimiento superiores al 80 % en las actividades semanales obligatorias (Caicedo & Vásquez, 2016, p. 51).

Esta mejora no puede explicarse exclusivamente por el entorno digital, sino por el rediseño pedagógico que dio sentido a cada actividad, conectándola con objetivos de aprendizaje claros, materiales accesibles y mecanismos de retroalimentación efectivos.

La participación no fue solo cuantitativa. El análisis cualitativo de los foros mostró un cambio en la calidad de la interacción. Los estudiantes no solo preguntaban, sino que respondían entre ellos, compartían estrategias de resolución, y reflexionaban sobre sus errores. Esta dinámica horizontal transformó el foro de un simple buzón de

dudas en un espacio colaborativo de construcción de conocimiento.

Tal como lo argumenta Vygotsky (1978), el aprendizaje se potencia en la interacción social, especialmente cuando se promueve entre pares, y el entorno Moodle ofreció condiciones adecuadas para ello.

En segundo lugar, se evidenció una mejora sustancial en la comprensión de los conceptos fundamentales del curso. Los resultados obtenidos en los cuestionarios automatizados y las tareas programadas mostraron un aumento sostenido en los puntajes promedio respecto a cohortes anteriores.

Más allá de las cifras, lo relevante fue la evolución progresiva del rendimiento: los estudiantes que partieron con dificultades evidentes lograron alcanzar niveles aceptables de dominio, y lo hicieron mediante una lógica de ensayo y error acompañada, en la que la retroalimentación oportuna jugó un papel decisivo. Como lo demuestran Black y Wiliam (2009), la evaluación formativa no solo mide, sino que orienta, reconstruye y sostiene el aprendizaje cuando es aplicada con sistematicidad.

Los testimonios recogidos por las autoras del estudio complementan la evidencia cuantitativa. Muchos estudiantes señalaron que, por primera vez, sentían que el curso estaba diseñado para que pudieran aprender, no para que quedaran por fuera.

Esta percepción subjetiva del entorno formativo — la sensación de que el proceso era posible, accesible,

justo— fue uno de los logros más significativos del piloto. En palabras de una estudiante: “No sabía nada de programación, pero no me sentí tonta. El curso me fue llevando paso a paso, y cuando me equivoqué, no me rajaron, me explicaron” (Caicedo & Vásquez, 2016, p. 52).

Finalmente, y de manera correlativa, se observó un impacto positivo en la permanencia estudiantil. La tasa de aprobación del curso, que históricamente no superaba el 35 %, se duplicó durante el semestre del piloto, superando el 70 %. Aunque la muestra no permite generalizaciones estadísticas, el resultado fue contundente como señal de que es posible transformar un curso históricamente expulsor en una experiencia formativa constructiva y retadora. Este logro cobra mayor relevancia si se considera el contexto estructural de exclusión en el que se inscribe la universidad y la alta vulnerabilidad social del estudiantado.

Los resultados del piloto, por tanto, validan la tesis central de este ensayo: repensar la enseñanza de programación desde un enfoque pedagógico crítico y tecnológicamente mediado no solo mejora los indicadores de rendimiento, sino que transforma la experiencia educativa en su sentido más profundo.

Con estrategias como esta, la participación se incrementa porque el estudiante se siente convocado; la comprensión mejora porque el contenido está secuenciado y mediado; y la permanencia aumenta porque el aula virtual deja de ser un espacio de evaluación punitiva para convertirse en un entorno de aprendizaje genuino.



## Capítulo 5. Enseñar programación en contexto: Tecno-pedagogía, enseñar sin reproducir exclusión

La experiencia de rediseñar el curso Fundamentos de Programación en la Universidad del Pacífico no puede ser comprendida como una simple mejora metodológica ni como un experimento tecnológico aislado. Más bien, se trata de un proceso formativo profundamente anclado en el reconocimiento del contexto territorial, social y epistémico de los estudiantes.

Enseñar programación en un territorio como Buenaventura —marcado por condiciones estructurales de exclusión, discontinuidades escolares, racismo sistémico y brechas digitales profundas— exige algo más que adaptar contenidos: requiere repensar de raíz la función de la educación técnica en un espacio históricamente negado.

En este sentido, la apuesta por un curso virtual mediado por Moodle 2.5 no se propuso replicar esquemas estandarizados de enseñanza ni trasladar contenidos de un aula presencial a una plataforma digital.

Se trató de construir, desde abajo y con intención pedagógica, una alternativa formativa situada, que asumiera la tecnología como medio y no como fin, y que se propusiera transformar una experiencia

históricamente expulsora en una oportunidad de aprendizaje significativa.

Este capítulo profundiza en tres dimensiones clave de ese proceso: la tecnopedagogía situada, la autonomía del estudiante como meta formativa, y la reconfiguración del rol docente en clave de mediación.

Uno de los principales errores en la implementación de tecnologías educativas en contextos vulnerables ha sido su naturalización acrítica. Como lo advierten Burbules y Callister (2000), pensar la tecnología como un conjunto de herramientas neutras, transferibles sin adaptación, implica desconocer su profunda carga ideológica y su capacidad para reproducir desigualdades.

En entornos como el del Pacífico colombiano — donde el acceso a conectividad es intermitente, los dispositivos son escasos y las trayectorias escolares están mediadas por múltiples formas de precariedad—, implementar una plataforma sin una lectura crítica del entorno puede no solo fracasar, sino reforzar las exclusiones que pretende resolver.

El rediseño del curso Fundamentos de Programación se constituyó precisamente como un ejercicio de tecnopedagogía situada. Esto significó reconocer que la tecnología, para ser educativa, debía adaptarse a las condiciones materiales, cognitivas y culturales del estudiantado.

No bastaba con instalar Moodle 2.5. Había que diseñar un entorno de aprendizaje que se acomodara a

los ritmos, tiempos, recursos y realidades de los estudiantes.

Esta adecuación se expresó, por ejemplo, en la elaboración de recursos de bajo peso descargable, el uso de videos breves compatibles con celulares básicos, la organización del curso en módulos autoaccesibles sin necesidad de conexión continua, y la creación de glosarios construidos con términos accesibles, relevantes y culturalmente cercanos.

El contenido disciplinar fue cuidadosamente mediado para no reproducir la lógica de la “transmisión vertical” del conocimiento, tan común en la enseñanza de disciplinas técnicas. Se incorporaron ejemplos locales, problemas contextualizados, y situaciones cotidianas que permitieran a los estudiantes reconocer la utilidad del pensamiento algorítmico en su propio entorno.

Este enfoque responde a lo que Freire (1997) denominó la “lectura del mundo”: el saber no se impone, se construye desde los códigos del oprimido, desde su experiencia vital, y en diálogo con saberes complejos.

Por eso, la tecnopedagogía situada no es solo un enfoque pragmático; es una posición política. Implica reconocer que las plataformas, los métodos, los ejemplos y las formas de evaluación no son neutros, y que si no se contextualizan, reproducen una forma sutil pero eficaz de exclusión: la de quienes se ven obligados a aprender en un lenguaje que no es el suyo, con herramientas que no dominan

y bajo lógicas que no los reconocen como sujetos válidos de conocimiento.

En esa medida, el rediseño del curso en Moodle 2.5 fue también una estrategia de justicia educativa, al demostrar que era posible enseñar programación de manera rigurosa pero adaptada, técnica pero comprensible, digital pero humana.

Esta forma de enseñar responde a lo que algunos autores han llamado pedagogía de la equidad digital (Warschauer & Matuchniak, 2010), es decir, una perspectiva que no se conforma con el acceso a la tecnología, sino que busca garantizar su uso significativo, formativo y empoderador. En este modelo, la mediación tecnológica no es un fin en sí misma, sino un instrumento para ampliar horizontes de posibilidad, democratizar el acceso al conocimiento computacional, y reconfigurar las condiciones de producción de saber desde los márgenes.

La experiencia de Buenaventura muestra, así, que es posible diseñar propuestas tecnopedagógicas sólidas en condiciones adversas, siempre que se parta de una comprensión ética y crítica del territorio.

Enseñar programación desde un enfoque situado no significa bajar el nivel académico, sino elevar la pertinencia pedagógica. No se trata de “acomodar” el curso, sino de reconstruirlo desde una lógica formativa que reconozca las desigualdades como punto de partida, y no como obstáculo insalvable.

## La autonomía del estudiante como meta formativa

En el rediseño pedagógico del curso Fundamentos de Programación, uno de los pilares estratégicos fue el desarrollo de la autonomía del estudiante como objetivo formativo central.

Esta decisión obedeció a un reconocimiento profundo de las condiciones del aprendizaje virtual y del tipo de competencias que exige la formación universitaria en el siglo XXI y no a una moda educativa ni a una delegación de responsabilidades desde el docente al estudiante.

La autonomía, en este marco, se constituye, no en una característica previa que se espera del estudiante, sino en una capacidad que debe ser construida, acompañada y valorada como parte integral del proceso de aprendizaje.

En entornos virtuales como Moodle 2.5, el estudiante se encuentra en un espacio que demanda mayores niveles de autorregulación, organización del tiempo, toma de decisiones, búsqueda activa de ayuda y seguimiento continuo de sus propios avances.

A diferencia de los cursos presenciales tradicionales, donde el ritmo lo marca el docente y el aula ofrece una estructura externa, en la virtualidad el marco de referencia es más difuso, y el riesgo de la desorientación o la procrastinación aumenta, especialmente para quienes no han tenido experiencias previas exitosas en este tipo de entornos.

Por eso, el curso no asumió la autonomía como condición de entrada, sino como meta formativa progresiva. Se diseñó una secuencia didáctica que ayudara al estudiante a transitar desde formas más dependientes de aprendizaje —basadas en explicaciones guiadas y tareas simples— hacia formas más autorreguladas, donde se promovía la exploración, la resolución autónoma de problemas, y la participación activa en espacios colaborativos.

Esta estrategia se apoyó en recursos como rúbricas de autoevaluación, retroalimentación inmediata en cuestionarios automatizados, calendarios claros y visibles en la plataforma, y actividades con plazos definidos pero flexibles.

Caicedo y Vásquez (2016) destacaron que muchos de los estudiantes que participaron en el piloto —aplicado en el segundo semestre de 2015— llegaron al curso con una baja percepción de autoeficacia, especialmente frente a asignaturas técnicas. Sin embargo, a medida que interactuaban con la plataforma y comprendían el funcionamiento de las herramientas, comenzaron a experimentar mejoras no solo en su rendimiento, sino también en su disposición hacia el aprendizaje. Varios reportaron sentir que “podían hacerlo solos”, que el curso estaba “pensado para ayudar, no para enredar” (p. 52), y que incluso cuando se equivocaban, tenían oportunidades para revisar, entender y corregir.

Este tipo de acompañamiento favoreció el desarrollo de la autonomía cognitiva y también de la autonomía emocional, entendida como la capacidad de

perseverar, de tolerar la frustración y de confiar en la posibilidad de aprender.

En este sentido, la autonomía no se construyó en soledad, sino dentro de una red de apoyos pedagógicos y tecnológicos cuidadosamente integrados en el entorno Moodle. Esta aproximación coincide con las propuestas de Zimmerman (2002), quien define el aprendizaje autorregulado como un proceso cíclico en el que los estudiantes planifican, monitorean y ajustan sus estrategias a partir de la retroalimentación recibida.

Además, el fomento de la autonomía fue coherente con un enfoque de educación centrado en el estudiante, que reconoce su papel activo en la construcción del conocimiento y que desplaza el énfasis desde la enseñanza hacia el aprendizaje.

Esta perspectiva ha sido ampliamente desarrollada en la pedagogía universitaria contemporánea, especialmente en el marco de la educación abierta y a distancia (Barberà, 2008), donde se reconoce que la autonomía no es una virtud individual aislada, sino una competencia compleja, relacional y situada.

En contextos como el de Buenaventura, fomentar la autonomía también tiene un carácter ético y político. Es una forma de resistir las representaciones deficitarias que asocian al estudiante de regiones periféricas con la dependencia, la falta de disciplina o la baja capacidad de autoorganización.

El curso rediseñado en Moodle 2.5 demostró que, con el acompañamiento adecuado, los estudiantes no solo pueden aprender programación, sino

transformar su relación con el saber y consigo mismos como aprendices. Aquí, la autonomía no fue solo un resultado colateral, sino una de las conquistas pedagógicas más significativas del proceso.

## **El rol docente reconfigurado en clave de mediación**

La virtualización de la enseñanza no solo transforma la relación del estudiante con el conocimiento, sino que exige una profunda reconfiguración del rol del docente.

Lejos de reducir su importancia, el paso a entornos digitales como Moodle 2.5 revela con claridad que la calidad de la experiencia educativa depende, en buena medida, de la capacidad del docente para asumir una función de mediador pedagógico: alguien que diseña entornos, anticipa dificultades, acompaña procesos y genera condiciones significativas de aprendizaje.

En el rediseño del curso Fundamentos de Programación, esta transformación del rol docente fue uno de los ejes más evidentes. La enseñanza dejó de centrarse en la transmisión directa de contenidos —propia del modelo tradicional basado en la exposición magistral y la evaluación sumativa— para dar paso a una figura docente que orquesta la experiencia formativa desde múltiples dimensiones: diseño instruccional, curaduría de contenidos, moderación de interacciones, retroalimentación

personalizada y seguimiento académico diferenciado.

Como sostienen Caicedo y Vásquez (2016), el docente del curso asumió el reto de convertirse en un “facilitador activo de los aprendizajes”, utilizando las herramientas de Moodle 2.5 no como fines en sí mismas, sino como medios para crear una red de mediaciones eficaces (pp. 47–52).

Esta reconversión coincide con lo planteado por Gilly Salmon (2000), quien describe la figura del moderador como un profesional que no solo domina los contenidos disciplinares, sino que estructura las fases de aprendizaje digital, estimula la participación, regula la interacción y ofrece acompañamiento emocional y cognitivo en entornos asincrónicos.

En este sentido, el docente no desaparece en la virtualidad, sino que se multiplica en funciones y se vuelve clave para sostener el proceso formativo, especialmente cuando el estudiante proviene de trayectorias educativas marcadas por el aislamiento, la autoexclusión o la desconfianza en su propia capacidad de aprender.

Uno de los aspectos más relevantes del nuevo rol fue la retroalimentación pedagógica constante. A través de los foros, los comentarios personalizados en tareas, las sugerencias ante errores frecuentes y los mensajes privados, el docente logró construir una presencia activa en la plataforma, que mitigó la sensación de soledad y generó confianza en el proceso.

Esta mediación fue más allá del cumplimiento de funciones: representó una forma de estar presente, de sostener y de generar un vínculo formativo en un entorno donde el riesgo de desenganche es permanente. Tal como lo afirman Laurillard (2012) y Coll (2013), el aprendizaje en línea requiere una interacción estructurada que no solo brinde contenido, sino que dialogue con el estudiante en su singularidad y en su proceso.

La planificación inicial del curso no fue una estructura rígida, sino una hipótesis de trabajo que se fue ajustando con base en las respuestas del grupo, en las dificultades identificadas, en los ritmos de avance y en las nuevas necesidades emergentes.

Esta flexibilidad exigió una disposición a escuchar, a revisar decisiones didácticas, a rehacer instrucciones y a reconfigurar estrategias de evaluación, lo cual evidencia un enfoque de enseñanza centrado en el estudiante y en su contexto, no en la comodidad del docente ni en las rutinas institucionales.

En un entorno como el del Pacífico colombiano, donde la distancia histórica entre instituciones educativas y comunidades ha generado desconfianza, la figura del docente mediador se vuelve clave para restaurar el vínculo educativo.

En lugar de reproducir una relación vertical, el nuevo rol docente apuesta por una relación horizontal, dialógica y empática, que reconoce al estudiante como sujeto activo, situado y valioso,

capaz de aprender programación si se le ofrecen las condiciones adecuadas.

Este enfoque está en sintonía con la pedagogía crítica de Freire (1997), que insiste en que enseñar no es transferir conocimiento, sino crear las condiciones para su construcción colectiva.

Finalmente, el docente como mediador en Moodle 2.5 también fue un agente de justicia educativa. Al identificar a tiempo a estudiantes en riesgo de deserción, al flexibilizar ritmos, al reconocer trayectorias particulares, y al diseñar actividades accesibles y retadoras a la vez, el equipo docente no solo enseñó programación: ejerció una práctica pedagógica transformadora, que hizo posible el aprendizaje en un territorio históricamente marginado.



## Capítulo 6. Moodle 2.5 desde Buenaventura: una propuesta replicable

La experiencia de rediseñar un curso crítico como Fundamentos de Programación, implementado a través de Moodle 2.5 en la Universidad del Pacífico, ofrece algo más que resultados locales. Representa la posibilidad de pensar una pedagogía digital transformadora desde los márgenes, capaz de desafiar las condiciones de precariedad estructural y producir aprendizajes significativos, sostenibles y replicables.

Buenaventura no fue un escenario incidental, sino un laboratorio pedagógico donde se puso a prueba el potencial de las tecnologías educativas cuando se alinean con criterios de justicia, pertinencia y contextualización.

Este capítulo explora lo que funcionó, lo que puede mejorarse y las claves para adaptar esta propuesta en otros contextos universitarios públicos.

### ¿Qué funcionó qué faltó, y qué puede mejorarse?

Uno de los logros más significativos del rediseño fue haber convertido una asignatura históricamente excluyente en una experiencia formativa viable, sin sacrificar el rigor técnico ni la profundidad conceptual. El curso no se “simplificó”, sino que se reconstruyó pedagógicamente, con una arquitectura

que favorecía la progresividad, la comprensión y la autonomía del estudiante.

Como está registrado Caicedo y Vásquez (2016), las tasas de participación, las calificaciones y la percepción del curso mejoraron sustancialmente durante el piloto implementado (pp. 50–52).

Otro elemento clave fue la adopción pedagógica crítica de Moodle 2.5, que dejó de usarse como repositorio de archivos para convertirse en un entorno activo de aprendizaje. Herramientas como foros, glosarios, cuestionarios automatizados y tareas con retroalimentación permitieron crear rutas formativas claras y accesibles.

Este uso pedagógicamente intencionado de un LMS que, si bien limitado frente a versiones posteriores, demostró ser suficiente cuando se combina con diseño instruccional riguroso y compromiso docente.

También funcionó la decisión de diseñar con base en las condiciones del territorio: materiales livianos, formatos compatibles con dispositivos básicos, secuencias adaptables, ejemplos contextualizados. Esta adecuación no solo facilitó el acceso, sino que permitió a los estudiantes verse reflejados en los contenidos, generando apropiación y motivación genuina. Tal como lo han señalado autores como Castells (2009), la apropiación tecnológica solo ocurre cuando los usuarios pueden conectar los entornos digitales con sus mundos de vida.

Sin embargo, la propuesta no estuvo exenta de limitaciones. Una de ellas fue la dependencia

excesiva del acompañamiento docente: muchos estudiantes necesitaban guía constante, lo que exigió un nivel de dedicación muy alto por parte del equipo docente.

Esto, si bien positivo en términos de mediación, también pone de relieve la necesidad de institucionalizar apoyos adicionales, como tutores pares, centros de asistencia técnica o módulos de inducción tecnológica.

Otro aspecto a mejorar fue la infraestructura tecnológica institucional. A pesar del esfuerzo por diseñar materiales accesibles, persistieron dificultades por caídas del sistema, lentitud en la carga de archivos y limitaciones en el soporte técnico. Esto sugiere que, aunque el diseño pedagógico es central, la sostenibilidad de la experiencia requiere inversión institucional en conectividad, servidores y acompañamiento especializado.

También se identificó la necesidad de reforzar las estrategias de evaluación de impacto a largo plazo. El piloto entregó resultados valiosos, pero para consolidar la propuesta como modelo replicable se requiere sistematizar más datos, analizar cohortes a mediano plazo, e incorporar herramientas de analítica de aprendizaje que permitan tomar decisiones basadas en evidencia.

## **Claves para la adaptación en otras universidades públicas**

La experiencia de Buenaventura no es un modelo cerrado ni un paquete replicable de forma

automática. Es, más bien, una propuesta pedagógica adaptable, cuyos principios pueden inspirar a otras universidades públicas que enfrentan desafíos similares: altas tasas de reprobación en cursos introductorios, contextos de vulnerabilidad socioeducativa, limitada apropiación tecnológica y tensiones entre currículo formal y realidad estudiantil.

Entre las claves identificadas, la primera es diseñar desde el contexto. Ninguna plataforma, por potente que sea, puede sustituir la lectura profunda del territorio.

Esto implica conocer al estudiantado, entender sus trayectorias, sus recursos, sus tiempos y sus formas de aprender. Moodle 2.5 demostró que incluso con herramientas básicas es posible construir entornos significativos si el diseño parte de la realidad y no de supuestos normativos.

La segunda clave es la formación docente en tecnopedagogía crítica. No basta con capacitar al profesorado en el uso técnico de la plataforma. Se requiere formación en diseño instruccional, en pedagogías activas, en evaluación formativa, en inclusión y en mediación digital.

El docente debe sentirse autor y no rehén de la tecnología. Como lo argumenta Coll (2013), enseñar con TIC exige “nuevas maneras de pensar la enseñanza y el aprendizaje, no solo nuevas herramientas”.

Una tercera clave es articular esfuerzos institucionales. La innovación pedagógica no puede

recaer únicamente en un grupo docente aislado. Se necesitan políticas universitarias que reconozcan, apoyen y escalen estas experiencias: inversiones en infraestructura, acompañamiento técnico, incentivos a la innovación, apoyo psicosocial al estudiantado y mecanismos de seguimiento académico.

Finalmente, el caso de Buenaventura enseña que las soluciones no siempre requieren la última versión del software, sino una convicción pedagógica clara, un compromiso territorial fuerte y una voluntad real de transformar la educación desde adentro. Moodle 2.5, en manos de docentes comprometidos, fue más potente que plataformas más avanzadas mal utilizadas. Esta lección es crucial en un momento en que muchas universidades invierten en tecnología sin garantizar su uso pedagógico pertinente.



## Capítulo 7. El ecosistema Moodle en 2025: de la versión 2.5 al LMS inteligente

La experiencia desarrollada en la Universidad del Pacífico con Moodle 2.5 demuestra que incluso con tecnologías modestas es posible transformar el aprendizaje cuando el enfoque pedagógico es claro, crítico y situado. Sin embargo, el panorama tecnológico actual plantea nuevos escenarios que merecen ser analizados con cuidado.

En la actualidad, los sistemas de gestión del aprendizaje (LMS) han evolucionado hacia entornos cada vez más complejos, interactivos e “inteligentes”, integrando herramientas de inteligencia artificial (IA), analíticas de aprendizaje, gamificación, accesibilidad universal y automatización de procesos formativos.

Moodle, en su versión más reciente (4.x), se ha adaptado a muchas de estas tendencias, ampliando su ecosistema funcional y reafirmando su compromiso con el software libre y la personalización institucional.

Este capítulo busca pensar el futuro sin perder de vista las lecciones del pasado reciente. Si bien es evidente que las plataformas han evolucionado, también lo es que el salto tecnológico no garantiza, por sí solo, una mejora en la calidad educativa.

La historia de Moodle 2.5 en Buenaventura nos advierte sobre los peligros de pensar que el cambio

viene “de afuera hacia adentro”, es decir, desde la interfaz o la funcionalidad técnica hacia la pedagogía. Por el contrario, el aprendizaje significativo requiere que la tecnología se articule a un proyecto formativo, que esté al servicio de la mediación docente y de la justicia educativa.

A partir de esta premisa, este capítulo se organiza en tres secciones. La primera analiza las tendencias actuales en los LMS y examina cómo Moodle ha incorporado herramientas como la inteligencia artificial, los sistemas de recomendación y las métricas de aprendizaje personalizado. La segunda realiza una comparación crítica entre Moodle, Canvas, Blackboard y Google Classroom, observando tanto sus fortalezas como sus límites desde una perspectiva pedagógica.

Finalmente, la tercera sección plantea una pregunta urgente: ¿es posible una tecnología educativa sin pedagogía? En tiempos de automatización y fetichismo digital, se vuelve necesario advertir sobre los riesgos de una innovación desanclada de los principios educativos fundamentales.

En conjunto, este capítulo no busca idealizar el futuro ni oponerse a él, sino proponer una lectura crítica y estratégica del ecosistema digital que habitamos y construimos. Porque si algo enseña el caso de Moodle 2.5 en Buenaventura, es que el cambio real comienza cuando la tecnología es reorientada por la pedagogía, y no al revés.

## Usabilidad, inteligencia artificial, analíticas de aprendizaje y gamificación

En la última década, los sistemas de gestión del aprendizaje (LMS) han experimentado una evolución significativa, tanto en su estructura técnica como en sus objetivos pedagógicos. Moodle, desde su versión 2.5 hasta la actual 4.4 (2025), ha pasado de ser una plataforma centrada en la administración de cursos a convertirse en un ecosistema de aprendizaje cada vez más inteligente, interoperable y centrado en la experiencia del usuario.

Este tránsito no ha estado exento de tensiones: si bien las nuevas funcionalidades ofrecen oportunidades para personalizar la enseñanza y acompañar el aprendizaje con mayor precisión, también plantean desafíos éticos, pedagógicos y políticos que deben ser abordados críticamente.

Uno de los avances más visibles ha sido la mejora en usabilidad. Moodle ha rediseñado su interfaz con principios de diseño centrado en el usuario, simplificando la navegación, organizando mejor los recursos, y permitiendo configuraciones más intuitivas tanto para estudiantes como para docentes. Este cambio responde a un reclamo histórico en la comunidad educativa: una plataforma no debería requerir formación técnica avanzada para ser utilizada con fines pedagógicos. En la medida en que la usabilidad mejora, la carga cognitiva asociada al entorno disminuye y se potencia la concentración en las tareas formativas (Norman, 2013).

Además, la incorporación progresiva de módulos de inteligencia artificial en Moodle ha abierto nuevas posibilidades para el acompañamiento personalizado. Plugins como MyLearning Analytics y Prediction Models permiten identificar patrones de comportamiento, predecir riesgos de deserción y ofrecer recomendaciones adaptadas al perfil de cada estudiante.

Estas herramientas, si bien poderosas, no están exentas de riesgos. Como advierten Ifenthaler y Yau (2020), los sistemas de IA pueden reforzar sesgos implícitos si no se diseñan con criterios éticos y pedagógicos claros. La tecnología no es neutral: si no se orienta desde la pedagogía crítica, puede derivar en mecanismos de vigilancia más que de cuidado.

En cuanto a las analíticas de aprendizaje, Moodle ha integrado reportes avanzados que permiten monitorear el progreso, la participación y el rendimiento de los estudiantes en tiempo real.

Estos datos, cuando se usan con criterio formativo, pueden enriquecer el acompañamiento docente, identificar dificultades emergentes y ajustar las estrategias pedagógicas. Sin embargo, si se utilizan únicamente con fines administrativos o punitivos, pueden alimentar una lógica de control más que de formación. El desafío está en pasar de la métrica al sentido pedagógico: que los datos sirvan para generar acciones de mejora contextualizadas, no para clasificar o excluir.

Otro de los desarrollos contemporáneos es la integración de elementos de gamificación, entendida como la incorporación de dinámicas lúdicas — recompensas, insignias, niveles, desafíos— con el fin de aumentar la motivación y el compromiso. Moodle ha permitido la creación de sistemas de logros personalizados, actividades interactivas y retroalimentación inmediata con elementos visuales atractivos.

Si bien la gamificación puede ser útil en cursos que presentan altos niveles de frustración o abandono, como Fundamentos de Programación, su efectividad depende de que no trivialice el contenido ni reemplace la profundidad del aprendizaje por estímulos superficiales. Como señala Deterding (2012), gamificar no es convertir el aula en un juego, sino diseñar experiencias de aprendizaje que generen sentido y conexión emocional con el saber.

En síntesis, el ecosistema Moodle en 2025 ha madurado tecnológicamente, pero esto no garantiza por sí solo una mejora educativa. La incorporación de IA, analíticas y gamificación representa una ampliación de los medios, no de los fines.

La clave sigue siendo la intencionalidad pedagógica con que estas herramientas son utilizadas. El riesgo, como se abordará más adelante, está en su uso acrítico, tecnocrático o descontextualizado. Por eso, toda innovación debe ir acompañada de formación docente, reflexión ética y diseño pedagógico situado. La tecnología puede ayudar, pero no reemplaza la pedagogía.

## **Comparación crítica con Canvas, Blackboard y Google Classroom**

En el panorama actual de la educación superior, los sistemas de gestión del aprendizaje (LMS) constituyen infraestructuras centrales del quehacer pedagógico institucional. Si bien Moodle continúa siendo una de las plataformas más utilizadas a nivel mundial —especialmente en instituciones públicas y proyectos educativos de código abierto—, también han ganado presencia otras soluciones como Canvas (Instructure), Blackboard Learn (Anthology) y Google Classroom (Google for Education).

Estas plataformas, aunque funcionalmente similares, expresan modelos pedagógicos, comerciales y filosóficos distintos, cuya comparación crítica permite afinar decisiones institucionales sobre tecnología educativa.

Moodle, en su concepción y evolución, ha sido un proyecto comprometido con el software libre, la personalización y la descentralización del conocimiento.

Su arquitectura modular, su comunidad global de desarrolladores y su compatibilidad con múltiples estándares lo convierten en un LMS altamente adaptable.

Sin embargo, esta misma flexibilidad ha supuesto históricamente una mayor curva de aprendizaje, tanto para docentes como para estudiantes. Hasta versiones recientes, la usabilidad y la estética de la interfaz representaban obstáculos importantes para la adopción masiva. No obstante, su versatilidad

permite una apropiación pedagógica más profunda, en especial en instituciones con necesidades particulares de diseño instruccional (Dougiamas, 2022).

Por su parte, Canvas se presenta como una alternativa moderna, limpia y altamente intuitiva. Su diseño de interfaz, basado en estándares de experiencia de usuario (UX), ha sido ampliamente elogiado por facilitar la navegación y reducir la carga cognitiva.

Canvas ofrece una integración nativa con múltiples aplicaciones externas (LTI), analíticas detalladas y herramientas de colaboración en tiempo real. Sin embargo, su modelo de negocio —basado en licencias propietarias y soporte centralizado— limita el nivel de personalización y genera una dependencia tecnológica externa, que puede ser problemática para universidades públicas con presupuestos restringidos o enfoques de soberanía tecnológica (Watson & Watson, 2007).

Blackboard Learn, uno de los pioneros del sector, combina robustez institucional con funciones avanzadas de gestión académica.

Su fortaleza histórica ha estado en el soporte a procesos administrativos complejos, como gestión de calificaciones, reportes institucionales y herramientas de evaluación formal. Sin embargo, ha sido objeto de críticas por su complejidad técnica, lentitud de interfaz y altos costos de licenciamiento. A pesar de su evolución reciente tras la fusión con Anthology, Blackboard sigue representando un

modelo corporativo cerrado, poco compatible con propuestas pedagógicas centradas en la autonomía docente o la adaptabilidad curricular (Morris & Stommel, 2017).

Por último, Google Classroom ha ganado popularidad por su sencillez y su rápida integración con el ecosistema Google (Drive, Meet, Docs, etc.). Su uso se disparó durante la pandemia por COVID-19, especialmente en instituciones con poca infraestructura previa.

No obstante, su simplicidad puede ser una debilidad en entornos universitarios más complejos: carece de herramientas avanzadas de evaluación, diseño instruccional progresivo o seguimiento pedagógico profundo. Además, al ser un producto gratuito de una corporación tecnológica con fines de lucro, genera dudas sobre la privacidad, la protección de datos y la dependencia de plataformas comerciales (Selwyn, 2016).

Desde una perspectiva crítica, entonces, la elección de un LMS no debe guiarse solo por criterios técnicos o estéticos. Lo que está en juego es el modelo educativo que se quiere consolidar: ¿un entorno abierto, adaptable y pedagógicamente personalizable, como Moodle? ¿Un ecosistema eficiente pero cerrado, como Canvas o Blackboard? ¿Una solución funcional y minimalista, como Google Classroom? La comparación no debe resolverse en términos de “mejor” o “peor”, sino de coherencia institucional, objetivos pedagógicos y condiciones de implementación.

En el caso colombiano, donde la mayoría de las universidades públicas trabajan con Moodle, el desafío no es reemplazarlo, sino potenciar su uso con criterio pedagógico.

La experiencia de Moodle 2.5 en Buenaventura demostró que la transformación educativa no depende de la herramienta, sino del sentido con que se la emplea. Una plataforma básica puede producir aprendizajes profundos si se articula con una propuesta didáctica robusta, mientras que una plataforma avanzada puede quedar reducida a una carpeta digital si no hay una mediación pedagógica clara.

## ¿Tecnología sin pedagogía?: riesgos y dilemas actuales

El avance vertiginoso de las tecnologías educativas ha abierto un abanico de posibilidades que, hasta hace pocos años, parecían utópicas: plataformas inteligentes capaces de adaptar contenidos a los estilos de aprendizaje, analíticas que predicen el riesgo de deserción, tutores virtuales impulsados por inteligencia artificial, y entornos gamificados diseñados para motivar y retener.

Sin embargo, este progreso técnico ha sido acompañado por un creciente desfase entre la sofisticación de las herramientas y la solidez de los marcos pedagógicos que las orientan.

Nos enfrentamos así a un dilema crucial para el presente y el futuro de la educación superior: ¿qué ocurre cuando la tecnología avanza más rápido que la reflexión pedagógica?

Este riesgo no es hipotético. Como lo han advertido críticos de la tecnología educativa como Selwyn (2016) y Williamson (2017), el despliegue acrítico de soluciones digitales puede terminar consolidando una pedagogía del control, de la eficiencia y de la vigilancia, más preocupada por medir que por comprender, más centrada en el rendimiento que en la formación.

Las plataformas que prometen personalización pueden convertirse en mecanismos de estandarización; los datos, en instrumentos de clasificación; y la automatización, en sustituto de la mediación humana.

En este escenario, el modelo de enseñanza corre el riesgo de reducirse a una lógica transaccional: contenidos precargados, actividades rutinarias, retroalimentación automática y acompañamiento mínimo. Esta tendencia, alimentada por discursos tecnocráticos de “optimización” y “escalabilidad”, debilita la dimensión relacional, dialógica y transformadora de la educación. Como señala Biesta (2010), enseñar no es simplemente facilitar el acceso a la información, sino crear las condiciones para la aparición del sujeto que aprende, en su singularidad, su agencia y su historicidad.

Otro dilema relevante es el de la colonización corporativa del entorno educativo. La proliferación de soluciones privadas, desarrolladas por grandes empresas tecnológicas bajo modelos cerrados y de monetización de datos, ha introducido lógicas de mercado en el corazón de la universidad.

Plataformas como Google Classroom o Blackboard, evidencian esta situación, aquí la educación deja de ser un bien común para convertirse en un segmento de negocio. Esto no solo vulnera principios de soberanía tecnológica, sino que erosiona el carácter público y democrático de la educación superior (Morozov, 2013).

En contextos como el colombiano, donde las brechas sociales y digitales persisten con fuerza, una tecnología sin pedagogía puede profundizar la exclusión en lugar de mitigarla. La implementación de herramientas avanzadas sin un acompañamiento docente adecuado, sin procesos de formación crítica y sin criterios de inclusión puede traducirse en un acceso aparente pero un aprendizaje precario.

La experiencia de Moodle 2.5 en Buenaventura muestra justo lo contrario: que la tecnología, cuando se articula con una pedagogía situada, puede ser emancipadora.

Por todo esto, más que entusiasmarse con la novedad técnica, las universidades deben preguntarse: ¿qué sentido pedagógico guía nuestras decisiones tecnológicas? ¿A quién sirven los datos que recolectamos? ¿Qué tipo de estudiante estamos formando cuando automatizamos la relación educativa? ¿Cómo protegemos el vínculo humano en medio de entornos cada vez más despersonalizados?

La respuesta no pasa por rechazar la tecnología, sino por subordinarla a una visión educativa humanista, crítica y ética. No se trata de volver atrás, sino de avanzar con lucidez. Una universidad que adopta

LMS inteligentes, IA y gamificación sin repensar su proyecto pedagógico corre el riesgo de reproducir un simulacro de innovación.

Por el contrario, una universidad que toma la experiencia de Moodle 2.5 como punto de partida, que escucha a sus docentes, que diseña desde el territorio y que forma comunidades de aprendizaje vivas, puede construir un futuro donde la tecnología esté al servicio de la equidad, el pensamiento crítico y la transformación social.

## Capítulo 8. Retos futuros para la enseñanza de programación en la era digital

A medida que las universidades avanzan en la transformación digital, la enseñanza de la programación se consolida como uno de los campos estratégicos de la formación profesional, no solo para ingenieros, sino para una ciudadanía cada vez más mediada por tecnologías computacionales.

Sin embargo, el porvenir de esta enseñanza no depende únicamente de avances técnicos o de plataformas más sofisticadas. Por el contrario, requiere abordar con seriedad un conjunto de retos estructurales que comprometen el sentido, la equidad y la sostenibilidad del aprendizaje en este campo.

Las brechas en rendimiento y permanencia, la desconexión entre currículo y contexto, la sobrecarga docente y la fragmentación entre innovación pedagógica y decisiones institucionales, no se resolverán por la vía de la actualización tecnológica. Más bien, exigen una reconfiguración integral del ecosistema educativo que vincule políticas públicas, marcos curriculares, mediaciones didácticas y condiciones socioculturales reales de los estudiantes.

Este capítulo propone mirar hacia el futuro sin perder de vista lo aprendido desde el margen. La experiencia de Moodle 2.5 en Buenaventura mostró que la transformación pedagógica es posible cuando

el territorio es reconocido, cuando el diseño instruccional es cuidadoso y cuando la tecnología se subordina al sentido pedagógico. Pero también mostró que esa transformación no puede descansar únicamente en el compromiso de un equipo docente: se necesitan políticas institucionales claras, formación continua, recursos sostenibles y una concepción crítica del aprendizaje.

Los tres apartados de este capítulo exploran dimensiones clave para proyectar una enseñanza de la programación más justa, pertinente y transformadora.

En primer lugar, se aborda la urgencia de fortalecer políticas institucionales, programas de formación docente y enfoques de diseño universal. En segundo lugar, se analiza el potencial de las microcredenciales y de nuevas estrategias de retención diferenciada. Finalmente, se plantea el tránsito necesario desde un aula técnica —orientada exclusivamente a la transmisión de contenidos— hacia un aula crítica que promueva pensamiento complejo, autonomía y compromiso ético.

## **Políticas institucionales, formación docente y diseño universal**

Pensar el futuro de la enseñanza de la programación en la universidad implica superar la lógica del proyecto aislado para avanzar hacia modelos institucionales integrados. La sostenibilidad y replicabilidad de experiencias pedagógicas transformadoras —como la desarrollada en Buenaventura con Moodle 2.5— dependen de su

inserción en marcos normativos, culturales y administrativos que las reconozcan, fortalezcan y expandan.

Sin políticas institucionales claras, la innovación corre el riesgo de agotarse en el entusiasmo individual o de diluirse con los cambios de personal, presupuesto o prioridad.

En este sentido, se requiere que las universidades adopten políticas que reconozcan la enseñanza de programación como un campo estratégico y transversal, tanto para la formación profesional como para la participación ciudadana. Estas políticas deben garantizar condiciones mínimas de infraestructura digital, acceso a plataformas, disponibilidad de recursos técnicos y humanos, y soporte psicosocial al estudiantado.

Pero también deben ir más allá de lo técnico y comprender que el desafío es pedagógico y político. Como sostiene Coll (2013), el verdadero reto de las tecnologías en la educación no está en su adopción, sino en su apropiación significativa.

Un segundo pilar de esta proyección es el fortalecimiento de la formación docente continua con enfoque reflexivo y situado. No basta con enseñar a los profesores a usar una herramienta: se requiere generar procesos formativos que integren diseño instruccional, evaluación formativa, acompañamiento emocional, mediación crítica y trabajo interdisciplinar.

El docente que enseña programación hoy necesita saber más que programar: necesita comprender

cómo aprenden sus estudiantes, qué miedos traen, qué trayectorias los marcan, y cómo construir experiencias formativas inclusivas. En esta línea, Tünnermann Bernheim (2019) ha enfatizado que la universidad del siglo XXI debe formar docentes que no solo instruyan, sino que medien sentidos, generen vínculos y transformen subjetividades.

El tercer elemento clave es la incorporación decidida del Diseño Universal para el Aprendizaje (DUA) como principio rector en la elaboración de cursos de programación. Lejos de ser una estrategia solo para estudiantes con discapacidad, el DUA propone una arquitectura curricular que anticipa la diversidad como norma, no como excepción. Esto implica ofrecer múltiples formas de representación de la información (textos, videos, esquemas, simulaciones), diversas formas de interacción (trabajo colaborativo, tareas individuales, foros), y formas flexibles de evaluación (proyectos, pruebas, autoevaluaciones).

En cursos como Fundamentos de Programación, donde la tasa de deserción es alta, diseñar desde la diversidad puede hacer la diferencia entre el fracaso recurrente y el aprendizaje significativo (CAST, 2018).

La implementación de estas líneas no solo permite mejorar el aprendizaje de los estudiantes actuales, sino que abre la puerta a nuevas poblaciones: estudiantes de zonas rurales, personas en reintegración social, jóvenes trabajadores, madres cabeza de hogar, comunidades indígenas y afrodescendientes. Es decir, democratiza el acceso al

conocimiento computacional y convierte la enseñanza de la programación en un proyecto de equidad social, no solo de productividad económica.

Por todo esto, las universidades que miren hacia el futuro deben hacerlo con una pregunta ética en el centro: ¿quién puede aprender programación y en qué condiciones?

Si la respuesta depende del acceso económico, la formación previa o la familiaridad con la tecnología, entonces estamos perpetuando una educación excluyente. Pero si diseñamos políticas, formaciones y cursos pensados para la diversidad, entonces estaremos construyendo una enseñanza de la programación más justa, más humana y más transformadora.

## **El potencial de las microcredenciales y estrategias de retención**

La enseñanza de la programación en la universidad enfrenta un doble reto: hacer que más estudiantes accedan, pero también que más estudiantes permanezcan y se gradúen. A pesar de los avances en virtualización, muchos cursos introductorios siguen funcionando como barreras estructurales que reproducen desigualdades previas.

En este escenario, la innovación pedagógica debe ir acompañada de estrategias institucionales de retención, articuladas con nuevas formas de certificación del aprendizaje que respondan a trayectorias diversas y no lineales.

Entre esas nuevas formas destaca el crecimiento de las microcredenciales: certificaciones académicas de corto alcance, enfocadas en competencias específicas y generalmente integrables en itinerarios de formación más amplios.

En el ámbito de la programación, las microcredenciales ofrecen una vía accesible y flexible para que estudiantes con distintos niveles de preparación, tiempo o disponibilidad puedan avanzar paso a paso, sin que el abandono de un curso implique necesariamente el abandono de todo el proyecto formativo.

Las microcredenciales pueden tener un efecto pedagógico y emocional importante. Para estudiantes que han enfrentado múltiples fracasos escolares, recibir un reconocimiento formal por el dominio de una competencia concreta —como diseñar un algoritmo básico o comprender estructuras condicionales— puede funcionar como un factor motivacional y de permanencia.

Como lo plantea Milligan y Kennedy (2017), estas certificaciones intermedias validan el proceso, no solo el producto, y permiten visibilizar el esfuerzo y el progreso en entornos de alta exigencia técnica.

En contextos como el de Buenaventura, donde muchos estudiantes de Ingeniería de Sistemas llegan con lagunas formativas profundas en matemáticas, lógica o manejo de herramientas digitales, el modelo tradicional de curso único, extenso y evaluado con un examen final, actúa como un filtro excluyente.

En cambio, una estructura modular basada en microcredenciales permitiría diseñar itinerarios formativos adaptativos, donde cada estudiante avanza a su ritmo, recibe retroalimentación continua y obtiene reconocimiento por cada paso logrado.

No se trata, sin embargo, de fragmentar la formación, sino de flexibilizarla sin perder rigor. Las microcredenciales deben estar diseñadas con criterios pedagógicos sólidos, alineadas con las competencias curriculares y articuladas en sistemas de reconocimiento institucional.

No basta con emitir "badges" digitales decorativos: se requiere una política clara de certificación por competencias, interoperabilidad entre plataformas y mecanismos de validación externa que garanticen su legitimidad (Oliver, 2019).

Paralelamente, las microcredenciales deben inscribirse en estrategias de retención diferenciada, que reconozcan la diversidad de causas que llevan a la deserción. No todos los estudiantes abandonan por falta de interés o capacidad: hay razones económicas, familiares, territoriales, culturales y de salud mental que inciden en los trayectos educativos. En este sentido, la retención debe entenderse como una responsabilidad colectiva de la institución, no como una carga individual del estudiante.

Algunas estrategias exitosas incluyen: sistemas de alerta temprana basados en analítica de aprendizaje, tutorías entre pares, acompañamiento

psicoeducativo, flexibilización de plazos, y diseño de actividades significativas que conecten el aprendizaje con los intereses del estudiante. Todas estas medidas, cuando se articulan con microcredenciales bien diseñadas, construyen un ecosistema formativo más accesible, humano y efectivo.

En definitiva, el potencial de las microcredenciales no reside únicamente en su valor como certificación, sino en su capacidad de reorganizar la experiencia educativa en torno al reconocimiento progresivo del aprendizaje, en lugar de la penalización del error o del desfase.

Integradas en estrategias de permanencia y acompañamiento, pueden convertirse en una herramienta poderosa para democratizar el acceso al conocimiento computacional en la universidad del siglo XXI.

## **Del aula técnica al aula crítica: reconfigurar la educación en ingeniería**

Durante décadas, la enseñanza de la programación y, en general, de las disciplinas de ingeniería ha estado fuertemente anclada a un modelo técnico-instrumental, centrado en la adquisición de habilidades operativas, la resolución de problemas lógicos, y la aplicación de conocimientos en contextos abstractos o descontextualizados.

Este enfoque, heredado de las tradiciones positivistas de la modernidad industrial, ha sido eficaz para formar profesionales altamente especializados, pero ha mostrado límites

importantes en su capacidad de responder a los desafíos sociales, éticos y culturales del presente.

La educación en ingeniería —y particularmente la enseñanza de la programación— no puede seguir reduciéndose a una transmisión de comandos, algoritmos y lenguajes. Hoy, cuando el código estructura crecientemente nuestras vidas (desde los sistemas de salud hasta los mecanismos de vigilancia, desde las plataformas de comunicación hasta las políticas públicas), es necesario formar ingenieros capaces de pensar críticamente el impacto de sus decisiones técnicas en la sociedad.

No se trata de renunciar al rigor técnico, sino de complementarlo con una pedagogía crítica que promueva la comprensión de las implicaciones humanas, políticas y ambientales de la tecnología.

Pasar del aula técnica al aula crítica implica, en primer lugar, reconfigurar el currículo para incluir perspectivas interdisciplinarias. Esto supone integrar contenidos sobre ética profesional, filosofía de la tecnología, pensamiento sistémico, historia de la informática, estudios sociales de la ciencia y análisis de datos con enfoque de derechos.

El objetivo no es añadir contenidos por fuera, sino repensar la programación como una práctica cultural, que responde a una lógica y a unos valores, y que debe ser puesta en cuestión por quienes la ejercen. Como sostiene Feenberg (2010), toda tecnología encarna formas de poder y decisión, y por eso educar en tecnología exige también educar en ciudadanía.

En segundo lugar, esta reconfiguración exige modificar las metodologías de enseñanza. No basta con enseñar a programar: hay que enseñar a aprender, a investigar, a cuestionar, a dialogar, a argumentar. En lugar de ejercicios cerrados con una única respuesta válida, el aula crítica propone proyectos abiertos, colaborativos, contextualizados y conectados con problemas reales del entorno.

Las tecnologías digitales, bien utilizadas, pueden ser aliadas en este proceso: simuladores, entornos virtuales, narrativas interactivas, estudios de caso, análisis de código abierto, entre otros. Pero la clave está en la intencionalidad pedagógica, no en el formato. Como afirma Giroux (2019), la educación crítica no ocurre por accidente: es fruto de una praxis consciente y comprometida.

El tránsito hacia el aula crítica también requiere redefinir el rol del docente. Más que transmisor de contenidos o evaluador de resultados, el profesor se convierte en un mediador que acompaña procesos, plantea preguntas, desafía certezas y construye comunidad.

Esto demanda tiempo, formación, apoyo institucional y un cambio cultural en las facultades de ingeniería, que tradicionalmente han privilegiado el rendimiento técnico sobre la reflexión pedagógica.

La experiencia en la Universidad del Pacífico, donde la programación fue enseñada desde un enfoque situado, reflexivo y adaptativo, demuestra que es posible abrir este camino incluso en condiciones

adversas, si se parte de una convicción ética y de una voluntad pedagógica clara.

Finalmente, pasar del aula técnica al aula crítica implica formar profesionales con conciencia social, sentido ético y compromiso territorial. En contextos como el del Pacífico colombiano, donde la tecnología ha llegado tarde o ha sido impuesta sin mediación, el ingeniero no puede ser un simple ejecutor: debe ser un constructor de soluciones contextualizadas, un puente entre el conocimiento técnico y las necesidades del territorio. La programación, en este marco, deja de ser una competencia individual para convertirse en una herramienta colectiva de transformación.



## Capítulo 9. Repensar la enseñanza de programación: síntesis y propuestas

### De la excepción a la estrategia: lecciones desde Buenaventura

El rediseño del curso Fundamentos de Programación en la Universidad del Pacífico no debe entenderse como un hecho afortunado e irrepetible, sino como una estrategia replicable que reveló el potencial transformador de una tecnopedagogía situada. Su relevancia radica precisamente en que tuvo lugar en un contexto estructuralmente desfavorecido: limitado en conectividad, con altos índices de deserción y bajo rendimiento, y con una historia acumulada de exclusión territorial.

Esa condición adversa no impidió la innovación, sino que la volvió urgente. La intervención pedagógica articulada a Moodle 2.5 operó no como una solución importada, sino como una respuesta diseñada desde el margen, con los estudiantes en el centro.

A diferencia de muchas iniciativas de educación virtual que fracasan por reproducir modelos homogéneos y descontextualizados, el caso de Buenaventura apostó por una ruta distinta: leer el territorio, entender a sus estudiantes, respetar sus trayectorias y diseñar con base en esa realidad.

En este sentido, se demostró que la innovación no es sinónimo de sofisticación técnica, sino de pertinencia pedagógica. Las lecciones que deja esta experiencia apuntan a una transformación posible, incluso cuando no se tienen todos los recursos, siempre que exista una voluntad ética y un compromiso institucional con el derecho a aprender. La verdadera enseñanza transformadora no se mide por el entorno ideal, sino por la capacidad de generar sentido donde antes había distancia o silencio.

### **Moodle como medio, no como fin**

La historia reciente de los entornos virtuales de aprendizaje ha estado marcada por una creciente sofisticación técnica que, sin embargo, no siempre se ha traducido en mejoras pedagógicas. La experiencia analizada en este ensayo muestra que Moodle 2.5, a pesar de ser una versión antigua, fue suficiente para diseñar un curso que no solo redujo la deserción, sino que cambió la relación de los estudiantes con la programación.

Este hecho obliga a revisar críticamente la lógica de la actualización tecnológica como sinónimo automático de innovación. No es la versión del LMS lo que garantiza aprendizajes significativos, sino el uso pedagógico que se hace de sus herramientas.

Moodle funcionó como un medio, no como un fin, porque se puso al servicio de un proyecto educativo anclado en el reconocimiento de la diversidad y en la construcción de itinerarios formativos accesibles, progresivos y culturalmente pertinentes. Convertir la plataforma en un espacio de encuentro

pedagógico requirió que el docente se apropiara del entorno virtual no para replicar la clase presencial, sino para transformarla.

Esta transformación consistió en diseñar actividades que partieran del error como oportunidad, en generar evaluaciones con sentido, en secuenciar contenidos de manera gradual, y en articular lo técnico con lo humano. Así, Moodle dejó de ser un simple contenedor de archivos y se convirtió en un espacio para el acompañamiento, la autonomía y el pensamiento crítico.

Reducir la virtualización a una operación administrativa o técnica es uno de los riesgos más graves de las universidades contemporáneas. En muchos casos, plataformas más potentes que Moodle no logran impacto educativo porque carecen de una intencionalidad pedagógica clara.

La enseñanza de la programación, como lo muestra esta experiencia, no necesita de entornos espectaculares, sino de mediaciones significativas. La clave está en recordar que toda herramienta, por útil que sea, debe ser guiada por un proyecto formativo que defina el para qué y el para quién del acto de enseñar.

## **Formación con sentido territorial, ético y transformador**

El futuro de la enseñanza de programación no puede seguir basado en modelos genéricos que desatienden las condiciones materiales, culturales y afectivas del estudiantado. Formar en programación es formar para intervenir en el mundo digital, pero

también para habitarlo críticamente. En este sentido, el proceso formativo debe integrar una ética pedagógica que reconozca la dignidad de los estudiantes, sus historias previas, sus condiciones de vida y sus formas diversas de aprender. La experiencia de Buenaventura fue pedagógicamente poderosa no porque se enseñara un lenguaje de programación más, sino porque ese lenguaje fue traducido a una pedagogía del reconocimiento.

Reconfigurar la enseñanza desde un sentido territorial implica abandonar la lógica de exportación de contenidos estandarizados y diseñar propuestas formativas que dialoguen con los referentes del territorio. En este diálogo, el docente juega un papel clave como mediador cultural, no como simple transmisor de conocimientos. Para ello, se requiere una formación docente que no solo fortalezca competencias tecnológicas o disciplinares, sino que forme en mediación, empatía, acompañamiento y diseño instruccional. El profesor debe ser capaz de leer el aula no como un espacio neutral, sino como un campo atravesado por tensiones sociales, emocionales y culturales que afectan profundamente el aprendizaje.

Pero el sentido ético y transformador también exige politizar la práctica docente: enseñar programación es enseñar poder. El código no es neutro. Decide qué es posible hacer, cómo se organiza la información, quién accede, quién queda fuera.

Es por eso que los ingenieros que formamos deben ser también sujetos críticos, capaces de comprender que su tarea no es solo resolver problemas técnicos,

sino contribuir a la construcción de un mundo más justo, accesible y habitable. Desde esta perspectiva, la enseñanza de la programación se convierte en una práctica profundamente política, en el mejor sentido del término.

## **Propuesta de lineamientos para una virtualización inclusiva y crítica**

A lo largo de este ensayo se han analizado múltiples dimensiones de lo que implica enseñar programación desde una perspectiva crítica, territorial y tecnopedagógica. A modo de síntesis, es posible identificar una serie de lineamientos que podrían orientar procesos similares en otras instituciones de educación superior pública:

Todo proceso de rediseño debe comenzar con un diagnóstico profundo del contexto. Conocer las condiciones de acceso, los perfiles de ingreso, las trayectorias escolares y las expectativas del estudiantado permite tomar decisiones pedagógicas ajustadas a la realidad. Sin este punto de partida, las innovaciones tienden a reproducir inequidades ya existentes.

El diseño instruccional debe asumir la diversidad como eje estructural, no como problema a resolver. Esto implica construir rutas formativas flexibles, escalonadas, con múltiples formas de acceso a los contenidos y alternativas de evaluación que no penalicen la diferencia. La programación puede enseñarse de forma progresiva si se reconoce que los estudiantes no parten del mismo lugar.

La plataforma tecnológica debe ser seleccionada y utilizada con base en su capacidad de favorecer la mediación pedagógica. Moodle demostró que no es necesario contar con la última versión del software, sino con una estructura que permita orientar, acompañar, retroalimentar y construir comunidad. Virtualizar no es digitalizar: es diseñar pedagogía en otro formato.

El rol docente debe ser fortalecido como pieza clave del proceso. Las universidades deben asumir la formación pedagógica de sus docentes como responsabilidad institucional, y no como una expectativa voluntarista. Los profesores necesitan espacios, tiempo, reconocimiento y acompañamiento para repensar su práctica.

La sostenibilidad de los cambios requiere su institucionalización. No basta con proyectos piloto exitosos: se necesitan políticas internas que documenten, difundan, evalúen y consoliden las innovaciones pedagógicas. Esto implica crear redes de intercambio, incentivar la sistematización de experiencias y establecer mecanismos de escalabilidad.

Finalmente, y como ya hemos explicado, toda propuesta debe orientarse desde una ética del cuidado. Enseñar programación en contextos vulnerables no es una tarea técnica, sino una apuesta política por democratizar el acceso al conocimiento y formar profesionales que no reproduzcan exclusión. Moodle 2.5, en Buenaventura, fue la prueba de que esto es posible. El reto ahora es hacerlo norma, y no excepción.



## Bibliografía

- Biesta, G. (2010). Good education in an age of measurement: Ethics, politics, democracy. Routledge.
- Burbules, N. C., & Callister, T. A. (2000). Watch IT: The risks and promises of information technologies for education. Westview Press.
- CAST. (2018). Universal Design for Learning Guidelines version 2.2. Center for Applied Special Technology. <https://udlguidelines.cast.org/>
- Caicedo, G. A., & Vásquez, N. D. (2016). Estrategias virtuales para fortalecer la enseñanza de Fundamentos de Programación en la Universidad del Pacífico. Universidad del Pacífico.
- Coll, C. (2013). Aprender y enseñar con las TIC: expectativas, realidad y potencialidades. Boletín de la Institución Libre de Enseñanza, 91, 15–32.
- Deterding, S. (2012). Gamification: Designing for motivation. Interactions, 19(4), 14–17. <https://doi.org/10.1145/2212877.2212883>
- Dougiamas, M. (2022). The Moodle philosophy and the future of open learning platforms. MoodleMoot Global.
- Feenberg, A. (2010). Between reason and experience: Essays in technology and modernity. MIT Press.
- Giroux, H. A. (2019). Pedagogía crítica contra el neoliberalismo. CLACSO.

- Ifenthaler, D., & Yau, J. Y. K. (2020). Utilising learning analytics to support study success in higher education: A systematic review. *Educational Technology Research and Development*, 68, 1961–1990. <https://doi.org/10.1007/s11423-020-09788-z>
- Milligan, C., & Kennedy, E. (2017). To badge or not to badge? A review of the literature. *Learning, Media and Technology*, 42(2), 177–190.
- Morris, S. M., & Stommel, J. (2017). A guide for resisting edtech: The case against Turnitin. Hybrid Pedagogy. <https://hybridpedagogy.org/resisting-edtech>
- Norman, D. A. (2013). *The design of everyday things* (Revised and expanded edition). MIT Press.
- Oliver, B. (2019). Making micro-credentials work for learners, employers and providers. Deakin University. <https://dteach.deakin.edu.au/wp-content/uploads/sites/103/2019/08/Making-micro-credentials-work.pdf>
- Selwyn, N. (2016). *Is technology good for education?* Polity Press.
- Tünnermann Bernheim, C. (2019). Educación superior en América Latina y el Caribe: diez años después de la CRES 2008. IESALC-UNESCO.
- UNESCO IESALC. (2022). *Tendencias de la educación superior en América Latina y el Caribe*. Instituto Internacional para la Educación Superior en América Latina y el Caribe.

- Watson, W. R., & Watson, S. L. (2007). An argument for clarity: What are learning management systems, what are they not, and what should they become? *TechTrends*, 51(2), 28–34. <https://doi.org/10.1007/s11528-007-0023-y>
- Williamson, B. (2017). *Big Data in education: The digital future of learning, policy and practice*. Sage.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.



## Las autoras

**Yowanna Karina Caicedo Guerrero**  
Ingeniera Telemática, especialista en Informática, Telemática y Administración de la Información, y Magíster en Gestión de Tecnología Educativa. Su labor se ha enfocado en el diseño de entornos virtuales de aprendizaje y la incorporación de TIC en la educación superior, con enfoque en innovación pedagógica y equidad regional.  
**ORCID:** 0009-0005-6805-5782

**Aura Liliana Vásquez Olaya**  
Ingeniera de Sistemas, especialista en Administración de la Información y Magíster en Gestión de Tecnología Educativa. Ha trabajado en la gestión académica de programas universitarios y en la formación docente en competencias TIC, contribuyendo al desarrollo de la educación virtual y mediada por tecnologías digitales.  
**ORCID:** 0009-0007-6570-7380

