ARTÍCULO ORIGINAL
10.23670/RC.2026.01.02

DIAGNÓSTICO DEL ENTORNO EDUCATIVO EN LA ENSEÑANZA DE PROGRAMACIÓN EN LA UAGRM

DIAGNOSTIC STUDY OF EDUCATIONAL ENVIRONMENT IN PROGRAMMING EDUCATION AT UAGRM

Leonardo Vargas Peña[1]
[1]Universidad Autónoma Gabriel René Moreno, Facultad de Ingeniería en Ciencias de la Computación y Telecomunicaciones.
***Correspondencia:** leonardovargas@uagrm.edu.bo. https://orcid.org/0009-0004-5905-4596

RESUMEN

El objetivo de este estudio fue diagnosticar el entorno educativo, la motivación estudiantil y el rendimiento académico en la enseñanza de programación para estudiantes de ingenierías no computacionales en la Facultad de Ciencias Exactas y Tecnología (FCET) de la Universidad Autónoma Gabriel René Moreno (UAGRM) durante el semestre 1-2025. Se realizó un estudio descriptivo, con enfoque mixto y diseño no experimental transversal. La población estuvo conformada por estudiantes matriculados en la asignatura MAT204/INF110 y los siete docentes que la impartieron. Se aplicó una encuesta a una muestra intencional de 23 estudiantes que ya habían aprobado la materia, entrevistas semiestructuradas a los siete docentes y observaciones no participantes en siete grupos. Los instrumentos fueron validados por juicio de expertos. Los datos cuantitativos se analizaron con estadística descriptiva y los cualitativos mediante análisis de contenido temático. Los resultados indican que los estudiantes perciben dificultades en la lógica de programación (media 2,74 en escala de Likert 1-5) y en la adaptación al estilo de aprendizaje personal (media 2,65). La motivación intrínseca es moderada (media 3,35). Los docentes coinciden en que la falta de recursos y personalización limita el aprendizaje. Se identificó una alta expectativa de los estudiantes por sistemas adaptativos (media 4,04). Se concluye que el fortalecimiento del entorno educativo requiere un modelo adaptativo basado en inteligencia artificial que vincule el desarrollo de la lógica y la aplicabilidad de la programación en el perfil profesional de cada ingeniería, con potencial impacto en los niveles de motivación y el rendimiento académico.

Palabras clave: Programación informática; Métodos de aprendizaje; Inteligencia artificial; Motivación; Evaluación del aprendiza

ABSTRACT

The objective of this study was to diagnose the educational environment, student motivation, and academic performance in programming education for non-computer engineering students at the Faculty of Exact Sciences and Technology (FCET) of the Universidad Autónoma Gabriel René Moreno (UAGRM) during the 1-2025 semester. An observational, descriptive study was conducted with a mixed approach and non-experimental cross-sectional design. The population consisted of students enrolled in the MAT204/INF110 course and the seven professors who taught it. A survey was applied to an intentional sample of 23 students who had already passed the subject, semi-structured interviews were conducted with the seven professors, and non-participant observations were carried out in seven groups. The instruments were validated by expert judgment. Quantitative data were analyzed using descriptive statistics and qualitative data through content analysis. The results indicate that students perceive difficulties in programming logic (mean 2.74 on a 1-5 Likert scale) and in adapting to their personal learning style (mean 2.65). Intrinsic motivation is moderate (mean 3.35). Professors agree that the lack of resources and personalization limits learning. A high expectation of students for adaptive systems was identified (mean 4.04). It is concluded that strengthening the learning environment requires an AI-based adaptive model that links the development of logic with the professional applicability of programming within each engineering field, with a potential impact on motivation levels and academic performance.

Keywords: Computer programming; Learning methods; Artificial intelligence; Motivation; Learning assessment.

Recepción: 05/11/2025 **Aceptación:** 31/03/2026

INTRODUCCIÓN

La enseñanza de la programación es un componente clave en la formación universitaria, especialmente en ingeniería, ya que desarrolla pensamiento computacional, resolución de problemas y razonamiento lógico (Wing, 2006). En el contexto de la transformación digital actual, las tecnologías emergentes están redefiniendo su papel en la educación superior (Haleem y otros, 2022).

Aun así, persisten dificultades en el aprendizaje de la programación, particularmente en la abstracción algorítmica y el dominio de estructuras de control (Watson & Li, 2014), incluso con apoyo tecnológico (Cheah, 2020). Revisiones sistemáticas (Luxton-Reilly y otros, 2018) confirman que estos desafíos en cursos introductorios siguen siendo un eje central de investigación en esta área.

El pensamiento computacional se ha posicionado como un marco clave para enfrentar estas dificultades, aunque su integración curricular implica retos pedagógicos (Hsu y otros, 2018). Investigaciones recientes destacan la necesidad de una enseñanza progresiva y contextualizada (Qian & Lehman, 2017), mientras que la educación en informática busca consolidar una visión unificada de la disciplina (Pears, 2002).

La pandemia de COVID19 aceleró la dependencia de las plataformas tecnológicas, obligando a las universidades a implementar enseñanza remota de emergencia a escala global (Bond y otros, 2014). Este despliegue repentino posicionó a los Sistemas de Gestión del Aprendizaje (LMS) en el eje de la continuidad académica, aunque la adopción forzada de sistemas como Blackboard y Moodle también generó importantes desafíos.

La investigación emergente de ese período reveló que, aunque los LMS eran esenciales, su eficacia estaba limitada por problemas de usabilidad, baja interactividad y una escasa alineación con las necesidades pedagógicas (Almusharraf, 2024). Estas limitaciones, desde interfaces poco intuitivas hasta la falta de adaptación a diversos estilos de aprendizaje, afectaron el compromiso estudiantil y evidenciaron las debilidades de un enfoque meramente transaccional, subrayando la necesidad de soluciones más adaptativas y centradas en el usuario.

En paralelo, el avance de la inteligencia artificial en educación ha abierto nuevas oportunidades para la personalización del aprendizaje, y las revisiones recientes destacan su potencial transformador en la educación superior (Zawacki-Richter y otros, 2019). Asimismo, se observa un crecimiento en el uso de técnicas inteligentes en entornos educativos (Chassignol y otros, 2018) , mientras que estudios actuales subrayan sus implicaciones pedagógicas para docentes y estudiantes (Meylani, 2024).

Los sistemas de aprendizaje adaptativo ajustan contenidos y retroalimentación según el desempeño individual, y las revisiones sobre soluciones impulsadas por IA en educación superior confirman su potencial efectividad (Meylani, 2024). También se han observado resultados prometedores en el uso de técnicas inteligentes en entornos colaborativos (Tan y otros, 2022), mientras que la analítica de aprendizaje destaca como herramienta estratégica para optimizar procesos institucionales académicos (Ifenthaler & Yau , 2020).

Asimismo, diversos estudios resaltan la importancia de las políticas institucionales en la adopción de analítica educativa (Tsai & Gasevic, 2017) , y análisis recientes del panorama en educación superior refuerzan su valor estratégico (Viberg y otros, 2018).

En América Latina, la integración tecnológica sigue siendo un factor clave para la calidad educativa en educación superior, aunque enfrenta desafíos vinculados a infraestructura, formación docente y adecuación pedagógica (Romero Saldarriaga y otros, 2024). En Bolivia, y particularmente en la Universidad Autónoma Gabriel René Moreno (UAGRM), no existen estudios que describan el proceso de enseñanza-aprendizaje de la programación en ingenierías no computacionales. En la Facultad de Ciencias Exactas y Tecnología (FCET), de las 12 carreras de ingeniería, cuatro presentan una situación heterogénea: tres solo abordan contenidos básicos o aplicados a métodos numéricos, y una (del área mecánica) no incluye ninguna asignatura de programación.

En contraste con la heterogeneidad de estas cuatro carreras, las otras ocho ingenierías de la FCET sí cuentan con un espacio formativo común: la asignatura MAT204/INF110 (Informática I/Introducción a la Informática), que constituye el primer contacto formal con la programación y los fundamentos del desarrollo de software. A pesar de su relevancia como núcleo articulador de la formación inicial en programación, aún se desconoce en profundidad cómo se desarrolla su proceso pedagógico, así como las percepciones, motivaciones y dificultades que experimentan docentes y estudiantes. En este escenario, resulta pertinente desarrollar un diagnóstico académico en carreras donde saber programar no constituye el eje central de formación profesional.

Problema de investigación: A pesar de la importancia de la programación en la formación ingenieril, aún falta un diagnóstico claro sobre el entorno educativo, la motivación estudiantil y el rendimiento académico en su enseñanza dentro de la FCETUAGRM. Esta brecha limita el diseño de intervenciones pedagógicas efectivas y dificulta mejorar la calidad formativa en esta disciplina crítica para la ingeniería.

Pregunta de investigación: ¿Cuáles son las características del entorno educativo, la motivación estudiantil y el rendimiento académico en la enseñanza de programación en la asignatura MAT204/INF110 de la Facultad de Ciencias Exactas y Tecnología de la UAGRM?

Objetivo general: Diagnosticar el entorno educativo, la motivación estudiantil y el rendimiento académico en la enseñanza de programación en la asignatura MAT204/INF110 de la FCET-UAGRM durante el semestre 1-2025, como línea base para la propuesta de un modelo adaptativo basado en inteligencia artificial.

Objetivos específicos: Describir las estrategias pedagógicas, los recursos tecnológicos y las prácticas de enseñanza empleadas por los docentes en la asignatura MAT204/INF110; Identificar los niveles de motivación de los estudiantes hacia el aprendizaje de la programación y su percepción sobre la utilidad de esta competencia; Determinar las principales dificultades de aprendizaje y de rendimiento académico en la asignatura; Explorar las expectativas de estudiantes y docentes sobre la personalización de la enseñanza y el uso de tecnologías adaptativas.

Estrategia metodológica: Para alcanzar estos objetivos, se diseñó un estudio con enfoque mixto que combinó encuestas a estudiantes, entrevistas a docentes y observaciones de clases. Esta triangulación metodológica permite obtener una visión integral y multidimensional del fenómeno estudiado.

Resultados esperados: Se espera identificar los factores principales que afectan el aprendizaje de la construcción de programas, así como las necesidades y expectativas de los actores involucrados. Estos hallazgos constituirán la base empírica para el diseño de un modelo adaptativo basado en inteligencia artificial que personalice los recursos y metodologías de enseñanza, con impacto en la motivación y el rendimiento académico.

Estructura del artículo: El documento se organiza en cinco secciones. Tras esta introducción, se presenta la sección de materiales y métodos, donde se detalla el diseño del estudio, la población y muestra, los instrumentos y el procedimiento. A continuación, se exponen los resultados organizados en tres subsecciones correspondientes a las técnicas empleadas. Luego, en la discusión, se triangulan los hallazgos y se comparan con la literatura existente. Finalmente, se presentan las conclusiones derivadas del diagnóstico.

MATERIALES Y MÉTODOS

Tipo de estudio y diseño

El estudio fue descriptivo, con enfoque mixto y un diseño no experimental de corte transversal, realizado durante el semestre 1-2025 para caracterizar el entorno educativo, la motivación y el rendimiento académico en la enseñanza de programación. Se llevó a cabo en la FCET de la UAGRM, enfocándose en MAT204/INF110, asignatura común en ocho de las doce carreras de ingeniería no computacional.

Además, esta investigación no implementa ningún sistema adaptativo, sino que cumple una función diagnóstica y descriptiva, orientada a generar evidencia empírica contextual que sirva como base para el diseño posterior de un modelo adaptativo institucional.

Población y muestra

a. Población:

- Estudiantes aprobados: 32 estudiantes de Programación II (INF210) que ya habían aprobado MAT204/INF110. Para la encuesta.
- Docentes: Los siete docentes responsables de la asignatura, tres de los cuales impartían dos grupos. Para la entrevista.
- Estudiantes cursantes: 300 estudiantes matriculados en MAT204/INF110 en el semestre 1-2025, distribuidos en 10 grupos. Para la observación directa.

b. Muestra:

- Para la encuesta: Se utilizó un muestreo intencional no probabilístico dirigido a estudiantes que hubieran aprobado MAT204/INF110 y que en 1-2025 cursaban INF210. De los 32 estudiantes que cumplían este criterio, 23 participaron voluntariamente (aproximadamente 72% de tasa de respuesta). La selección se centró en este subgrupo porque permite obtener percepciones retrospectivas completas sobre la experiencia en la asignatura, y la alta tasa de participación asegura una representatividad adecuada de este perfil estudiantil.
- Para las entrevistas: Se incluyó a la totalidad de los siete docentes, garantizando la representación de todos los enfoques y estilos pedagógicos presentes en la asignatura.
- Para las observaciones: Se observaron siete de los 10 grupos de MAT204/INF110, asegurando que cada docente fuera observado al menos en uno de sus grupos (cuando tenían dos, se seleccionó uno). Esto permitió capturar la diversidad de dinámicas de enseñanza en la asignatura.

Variables de estudio

El estudio considera tres variables principales, definidas conceptualmente y operacionalizadas según se presenta en la Tabla 1.

a. Definición Conceptual de las Variables

- Entorno Educativo en la Enseñanza de Programación: Comprende las estrategias pedagógicas empleadas por los docentes, el uso de recursos tecnológicos y el grado de adaptación de la enseñanza a las necesidades del estudiante, así como la percepción de dificultad de los contenidos.
- Motivación hacia la Programación: Se refiere al interés, la valoración de la utilidad y la autoeficacia percibida por los estudiantes respecto al aprendizaje de programación, factores asociados a su compromiso y persistencia.
- Rendimiento Académico en Programación: Incluye el nivel de logro del estudiante según su historial de aprobación, su autoevaluación del desempeño y las dificultades reportadas, complementado con la percepción docente sobre el rendimiento general

b. Definición Operacional de las Variables

- Entorno Educativo en la Enseñanza de Programación: Se operacionalizó mediante la observación directa de las prácticas de enseñanza, las respuestas de los docentes estrategias y uso de recursos, y las percepciones de los estudiantes sobre la adaptación de la enseñanza y la dificultad de los conceptos.
- Motivación hacia la Programación: Se operacionalizó a través de las respuestas de los estudiantes en la encuesta, específicamente en relación con su interés, la utilidad que perciben de la programación para su carrera, y su nivel de esfuerzo como un indicador de su confianza en sus capacidades.
- Rendimiento Académico en Programación: Se operacionalizó a través de los reportes de los estudiantes sobre su historial de aprobación, sus percepciones sobre las dificultades de aprendizaje como indicadores inversos de rendimiento, y la perspectiva de los docentes sobre las dificultades y el rendimiento de estos.

Tabla 1. Operacionalización de Variables

Variable	Dimensión	Indicador	Instrumento
Entorno educativo (Variable Independiente)	Estrategias de enseñanza	Predominancia de exposición magistral	Guia de Observación (Categórica: diferentes rangos de categorías)
		Frecuencia de resolución de problemas/prácticas	Entrevista docente (Abierta)
	Personalización actual	Percepción docente de ajuste al ritmo del estudiante	Entrevista docente (Abierta)
		Percepción estudiantil de adaptación pedagógica	Encuesta estudiante (Likert 1-5)
Motivación (Variable dependiente)	Interés y valoración	Interés por aprender programación	Encuesta estudiante (Likert 1-5)
		Percepción de utilidad en la carrera	Encuesta estudiante (Likert 1-5)
	Persistencia y dedicación	Disposición a continuar ante dificultades	Encuesta estudiante (Likert 1-5)
		Tiempo dedicado a prácticas fuera de clase	Encuesta estudiante (Likert 1-5)
Rendimiento (Variable dependiente)	Dificultad conceptual percibida	Comprensión de lógica de programación	Encuesta estudiante (Likert 1-5)
		Relación teoría–práctica	Encuesta estudiante (Likert 1-5)
	Historial de aprobación	Aprobación previa de asignaturas del área	Encuesta estudiante (Likert 1-5)
		Autoevaluación del desempeño general	Encuesta estudiante (Likert 1-5)

Técnicas e instrumentos de recolección de datos

Se emplearon tres técnicas con sus correspondientes instrumentos, diseñados específicamente para este estudio y validados mediante juicio de expertos.

a. Encuesta a estudiantes

- Técnica: Cuestionario autoadministrado en línea (Google Forms) vía WhatsApp.

- Instrumento: Cuestionario estructurado de 26 ítems organizados en seis secciones: datos generales, sociodemográficos y académicos (8 preguntas semi cerradas); motivación e interés por la programación (4 ítems en escala Likert 1–5); dificultades de aprendizaje (4 ítems Likert); entorno educativo y metodología docente (4 ítems Likert); expectativas de personalización (4 ítems Likert); dos preguntas abiertas sobre dificultades y sugerencias.

- Validación: Revisión de tres expertos en educación, metodología e informática, quienes recomendaron ajustes de redacción y agregaron dos ítems (las preguntas abiertas). Se realizó una prueba piloto con cinco estudiantes similares a la muestra para verificar comprensión y ajustar escalas.

b. Entrevista a docentes

- Técnica: Entrevista semiestructurada presencial (fotocopias).
- Instrumento: Guía con 23 preguntas, distribuidas en: datos generales, sociodemográficos y académicos (cinco preguntas semi cerradas); tres secciones con 18 preguntas abiertas (cinco preguntas generales, ocho preguntas abiertas sobre experiencia docente y cinco preguntas sobre enseñanza personalizada).
- Validación: Revisión de los mismos tres expertos. Se realizó una entrevista piloto con un docente externo a la muestra para afinar la formulación.

c. Observación no participante

- Técnica: Observación sistemática no participante (fotocopias).
- Instrumento: Guía estructurada con 24 ítems distribuidos en seis secciones: datos generales (7 ítems), entorno de aprendizaje (3 ítems), interacción docente-estudiante (3 ítems), participación estudiantil (4 ítems), estrategias de enseñanza (4 ítems), adaptación del aprendizaje (3 ítems), combinando escalas categóricas y registros descriptivos dependiendo del ítem.
- Validación: Revisión por los mismos expertos y observación piloto en un grupo no incluido, útil para ajustar criterios y calibrar la forma de registro.

d. Procedimiento

La recolección de datos se realizó entre las semanas 3 y 10 del semestre 1-2025 siguiendo el siguiente proceso:

- Semana 3-4: Solicitud de permisos institucionales (vicedecano FECT) y con los docentes para explicar los objetivos y programar entrevistas y observaciones.
- Semanas 5-6: Realización de entrevistas semiestructuradas a los docentes (≈30 minutos cada una), las cuales fueron grabadas y transcritas con su autorización.

- Semana 7-8: Aplicación de la encuesta mediante un formulario en línea enviado por WhatsApp, disponible durante siete días y acompañado de dos recordatorios.

- Semanas 9-10: Observaciones de clase en los grupos seleccionados, asistiendo a una sesión completa por grupo y registrando la información en la guía correspondiente sin intervenir en la dinámica.

Todos los participantes fueron informados previamente sobre el propósito del estudio y su participación voluntaria, otorgando su consentimiento informado mediante casilla de aceptación en la fotocopia o el formulario en línea respectivo.

Análisis de datos

a. Datos cuantitativos: Las respuestas de la encuesta fueron exportadas a Microsoft Excel y analizadas mediante estadística descriptiva. Se calcularon frecuencias y porcentajes para variables categóricas (edad, género, nota, número de intentos), así como medias y recuentos para los ítems en escala Likert. Los resultados se presentaron mediante tablas y gráficos para facilitar su interpretación.

b. Datos cualitativos: Las entrevistas transcritas y las respuestas abiertas se analizaron mediante análisis de contenido temático, siguiendo tres etapas:

   - Preparación y organización de los datos: Lectura completa del material cualitativo para familiarización y comprensión global.

   - Identificación y categorización: Detección de patrones recurrentes y agrupación en categorías temáticas (p.ej., lógica de programación, aplicación práctica, gestión del tiempo).

   - Síntesis interpretativa: Interpretación personal de los hallazgos y selección de fragmentos textuales representativos que ilustran y validan cada patrón recurrente con el apoyo de herramientas ofimáticas para la organización de las ideas.

c. Consideraciones éticas: La participación en el estudio fue voluntaria y aprobada por los participantes. Se garantizó el anonimato mediante el uso de códigos alfanuméricos: D1–D7 para docentes, G1–G7 para grupos observados y E1–E23 para estudiantes encuestados. En los resultados no se divulgaron nombres. Asimismo, los datos fueron almacenados de forma segura en un equipo con acceso restringido exclusivamente al investigador.

RESULTADOS

La siguiente etapa del diagnóstico consiste en el análisis exhaustivo de la información obtenida. El objetivo de este análisis es identificar los patrones, tendencias y aspectos clave relacionados con el entorno educativo actual, la motivación de los estudiantes hacia la programación y su rendimiento académico en la asignatura MAT204/INF110.

Resultados de las entrevistas a docentes

Se entrevistó a los siete docentes que impartían MAT204/INF110 durante el semestre 1-2025. Sus edades oscilaron entre 41 y 80 años (media = 57,86 años) y su experiencia entre 4 y 40 años (media = 20,14 años). Cinco eran varones (71%) y dos mujeres (29%). A partir del análisis temático, emergieron las siguientes categorías principales:

- Dificultades conceptuales de los estudiantes: La dificultad más mencionada fue la lógica de programación (cinco, 71,4%), descrita como la incapacidad para "pensar y ordenar las soluciones". Tres eligieron a los problemas con diagramas, comandos y sintaxis; además se reportó la falta de base matemática, el análisis de programas. Coinciden en que los estudiantes presentan debilidades en el razonamiento secuencial y en la comprensión de estructuras fundamentales.

- Estrategias pedagógicas empleadas: Predominan estrategias como la resolución de ejercicios prácticos, el uso de ejemplos cotidianos, ambas con dos. Otros recursos son la introducción mediante pseudocódigo (PSeInt), talleres diagnósticos, uso de Excel para reforzar razonamiento matemático y atención individualizada ("insistir uno por uno"). Las estrategias tienden a combinar procedimientos estructurados con actividades prácticas guiadas.

- Uso de tecnología educativa: Todos utilizan Moodle para subir materiales y actividades. Cuatro usan videos de YouTube, mientras que dos emplean Google Classroom o Google Meet para apoyo adicional. También se usa WhatsApp como medio de comunicación. El uso de tecnología es consistente, aunque no necesariamente orientado a la personalización.

- Desafíos principales en la enseñanza: Los desafíos se agrupan en dos ejes: Desarrollar la lógica de programación (cuatro, 57,1%), destacando la dificultad de "hacerles entender la secuencia lógica de los algoritmos". Motivar y mostrar aplicabilidad (tres, 42,9%), señalando la importancia de evidenciar que la programación es una "herramienta de trabajo" útil para sus carreras. Este patrón coincide con lo observado en las dificultades conceptuales.

- Motivación estudiantil: La motivación es percibida como moderada o baja. Dos indicaron que los estudiantes no perciben la utilidad inmediata de la programación en sus carreras. Se observó que la motivación aumenta cuando logran visualizar resultados concretos, también se mencionó que la motivación depende de la carrera, pues algunas muestran mayor disposición que otras.

- Limitaciones de recursos e infraestructura: Cinco señalaron problemas como insuficiencia de equipos, laboratorios desactualizados, falta de mantenimiento e internet deficiente. Además de otras relacionadas con la formación matemática del estudiantado y la falta de capacitación docente en nuevas tecnologías.

- Conocimiento y disposición hacia metodologías adaptativas e inteligencia artificial: Tres señalaron no conocer o utilizar las metodologías adaptativas, dos indicaron conocerlas, pero con uso esporádico, y dos mencionaron limitaciones para su aplicación. Cinco expresaron disposición a participar en un programa piloto con IA. Se perciben a estas herramientas como "una ayuda" o "muy útiles", aunque reconocen barreras para su uso (falta de software y escasa formación previa).

- Percepciones sobre la enseñanza personalizada y adaptativa: Seis implementaron algún tipo de adaptación, de forma empírica: variación en ritmos, múltiples formatos de contenido o atención diferenciada según necesidad. Los más valorados para personalizar son: la participación (cinco) y los resultados de evaluaciones (cuatro). Los obstáculos fueron: falta de plataformas adecuadas (dos), ausencia de formación docente (dos), falta de tiempo por grupos numerosos y limitaciones en la base académica y motivación de los estudiantes (dos).

- Recomendaciones para mejorar la enseñanza: Se concentraron en tres áreas: cuatro indicaron la infraestructura tecnológica (instalar y mantener laboratorios, actualizar equipos y software. Con dos docentes cada una: la formación docente (en nuevas tecnologías y metodologías) y la revisión curricular (reorganizar asignaturas previas y ajustar contenidos según carrera). También sugirieron incorporar "nuevos métodos de enseñanza" y realizar mayor seguimiento.

Resultados de la encuesta a estudiantes

Se encuestó a 23 estudiantes de Programación II (INF210), todos de Ingeniería Electrónica, que ya habían aprobado MAT204/INF110. La edad osciló entre 20 y 38 años (media = 23), con predominio masculino: 21 hombres (91%) y dos mujeres (9%). La nota media de aprobación fue 63,61, y el 78,26% aprobó en el primer intento, el resto necesitó dos intentos, lo que indica que una parte del grupo enfrentó dificultades iniciales en programación.

Por un lado, las preguntas cerradas se centraron en las cuatro categorías principales de análisis, cada categoría tenía cuatro preguntas, utilizando una escala de Likert de 1-5 (1 = Totalmente en desacuerdo, 5 = Totalmente de acuerdo):

- Motivación e interés por la programación: Obtuvo valores moderados. El interés intrínseco obtuvo una media de 3,35, la percepción de utilidad para la carrera fue 3,33, y la disposición al esfuerzo alcanzó 3,17. El valor más alto fue la utilidad futura profesional con 4,00, indicando que reconocen la importancia de la programación a largo plazo (ver gráfico 1).

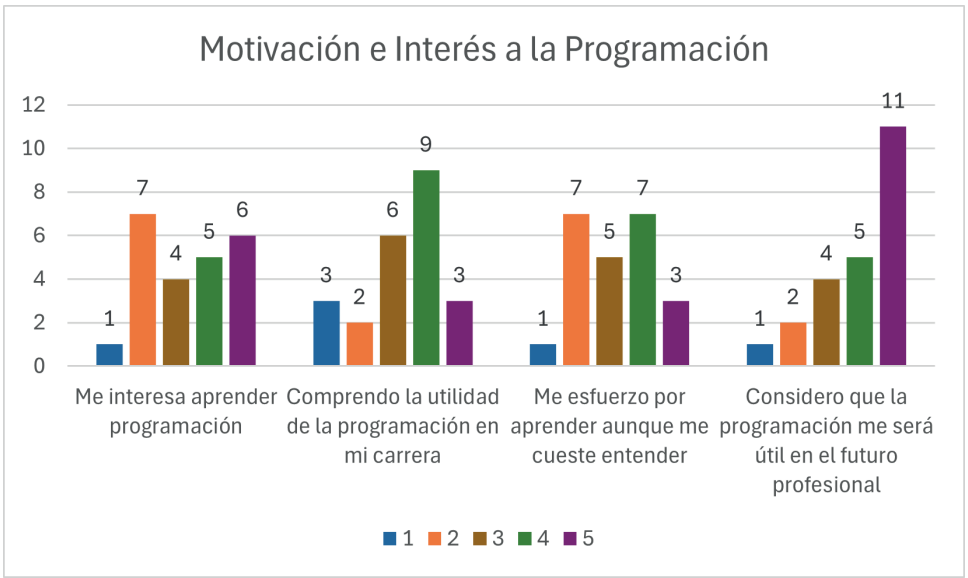


Gráfico 1. Motivación e Interés a la Programación

- Dificultades de aprendizaje: Principalmente la lógica de programación (media 2,74) y la relación teoría–práctica (2,61). Percepciones relativamente bajas sobre la adaptación de la enseñanza (2,65) y el ritmo de avance (2,43), lo que sugiere que las clases no se ajustan a los procesos de aprendizaje (ver gráfico 2).

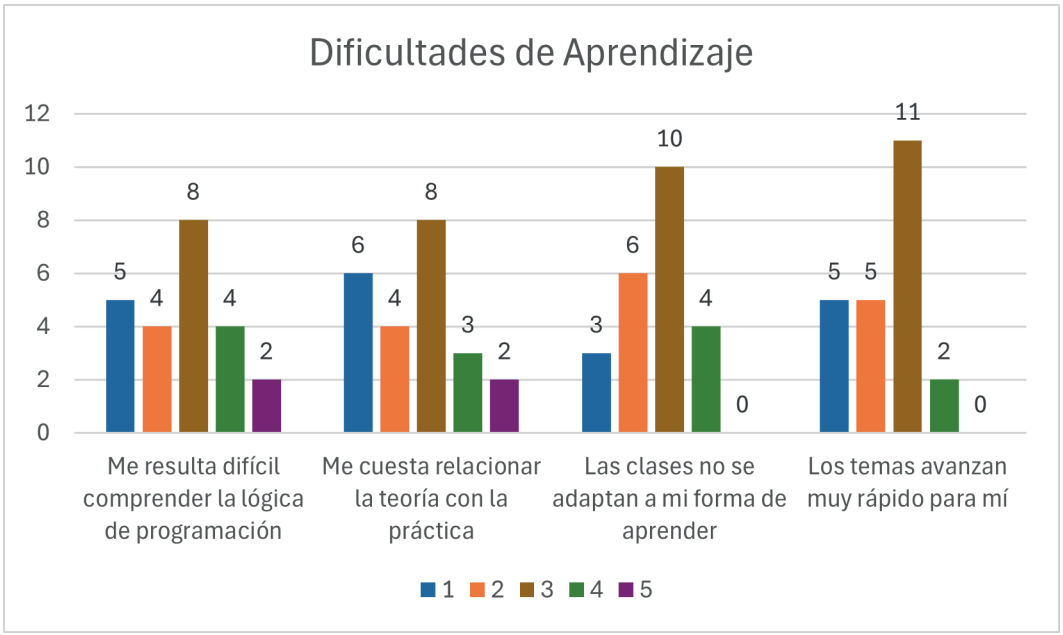


Gráfico 2. Dificultades de Aprendizaje

- Recursos y metodología docente: Oportunidades de mejora en el entorno educativo: el uso de herramientas adecuadas y la suficiencia del material fueron evaluados con media de 2,48. La integración de ejemplos obtuvo 2,35. El enfoque docente fue calificado de manera neutral con 2,57 (ver gráfico 3).

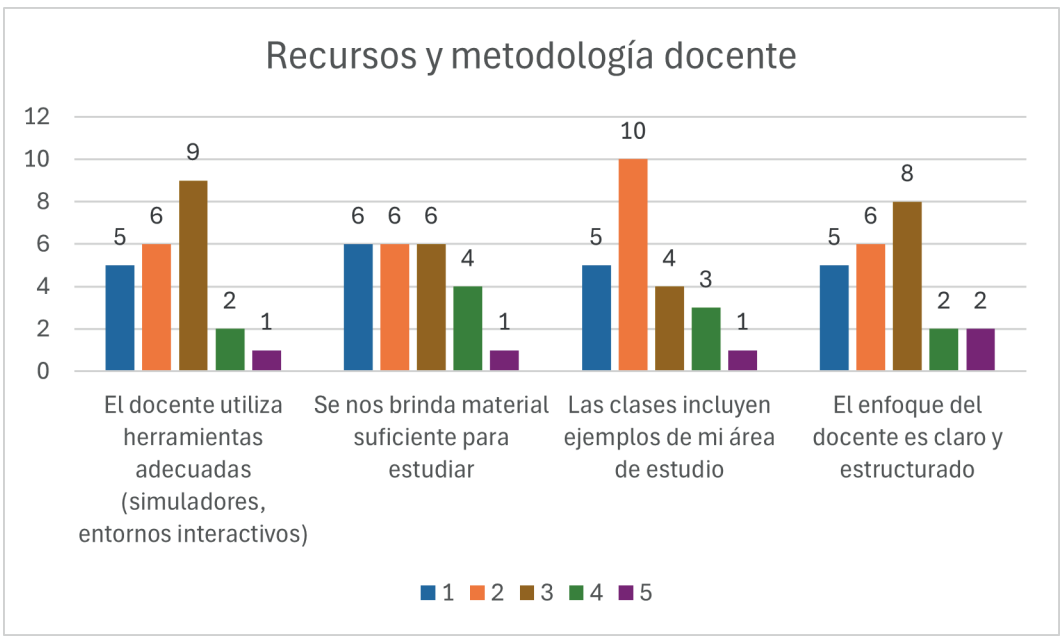


Gráfico 3. Recursos y metodología docente

- Expectativas sobre personalización: Fueron altas y son un hallazgo crucial: la adaptación al ritmo individual y los ejercicios personalizados (con media de 3,57), los sistemas que reconozcan avances y debilidades (3,74) y la fuerte disposición a usar plataformas inteligentes (4,04). Esto evidencia una fuerte apertura hacia sistemas adaptativos con IA (ver gráfico 4).

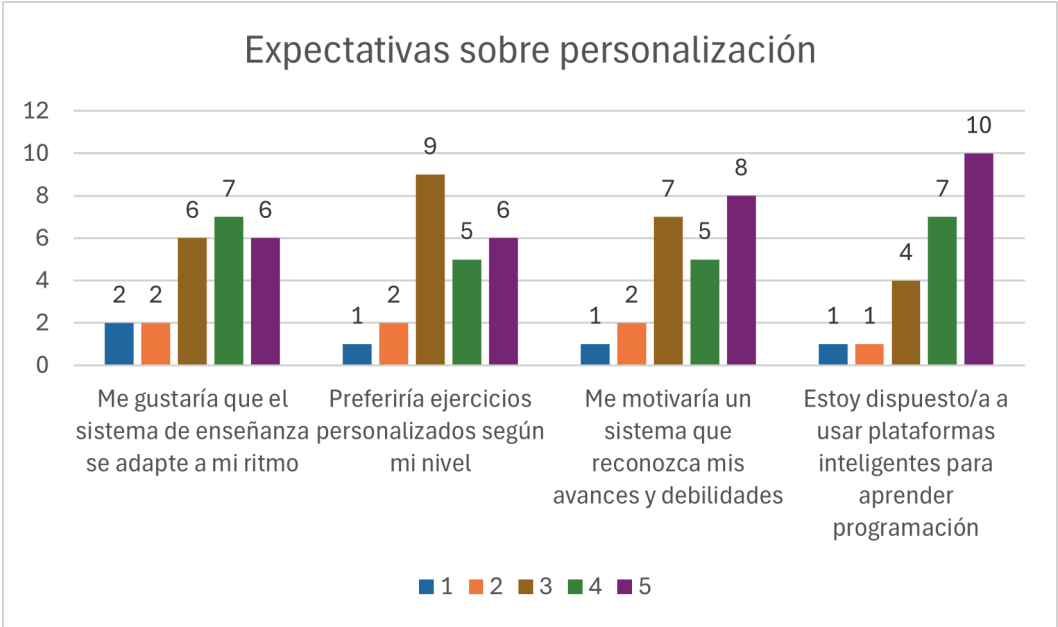


Gráfico 4. Expectativas sobre personalización

Por otro lado, las preguntas abiertas se centran en conocer las opiniones de los estudiantes sobre las dificultades más frecuentes al aprender programación y sus sugerencias para mejorar las clases de programación:

- Dificultades más frecuentes al aprender programación: Las dificultades se agruparon en cuatro categorías:

  ✓ Lógica de programación – mencionada por 11 estudiantes (47,8%): Señalaron dificultades para "entender el avance lógico del algoritmo" o "crear la estructura del algoritmo".

  ✓ Sintaxis, comandos y lenguajes – 6 estudiantes (26,1%): Indicaron problemas para recordar palabras clave, símbolos y cambios entre lenguajes.

  ✓ Tiempo y recursos – 3 estudiantes (13,0%): Incluyen falta de tiempo para estudiar o limitaciones por trabajo y economía.

- ✓ Interpretación de problemas y conceptos – 2 estudiantes (8,7%): Reportaron dificultades para traducir ejercicios a pasos programables.

- Sugerencias para mejorar las clases de programación: Las recomendaciones se clasificaron en cinco categorías:

  - ✓ Mejora de infraestructura tecnológica – 8 menciones (34,8%): Solicitan actualizar computadoras, mejorar laboratorios y contar con materiales como Arduino.
  - ✓ Más práctica y ejemplos aplicados a su carrera – 7 menciones (30,4%): Piden ejercicios contextualizados y clases más prácticas.
  - ✓ Acompañamiento y apoyo docente – 5 menciones (21,7%): Proponen auxiliaturas, ayudantes y sesiones de práctica con el docente.
  - ✓ Herramientas tecnológicas e IA – 4 menciones (17,4%): Incluyen sugerencias de IA, plataformas digitales o cursos sobre prompt engineering.
  - ✓ Aspectos metodológicos y didácticos – 4 menciones (17,4%): Solicitan más dinamismo en clases y evaluaciones, así como mayor enfoque en lógica.

Resultados de la Observación

Se realizaron observaciones no participantes en siete grupos (uno por cada docente). La Tabla 2 resume los hallazgos principales:

Tabla 2. Síntesis de las observaciones de clases

Aspecto observado	Descripción
Tipo de aula	2 grupos en aula tradicional, 2 en laboratorio, 3 mixto
Recursos disponibles	Pizarra en todos; proyector en 5; laboratorio con 80% de equipos funcionales
Metodología predominante	Clase magistral con preguntas dirigidas (5 grupos); trabajo individual supervisado (5 grupos)
Retroalimentación profesor	Moderada (media 3,5/5); general (5 grupos), específica (2 grupos)
Participación estudiantil	Moderada (media 3,2/5); preguntas esporádicas, distracciones con dispositivos móviles
Preguntas Frecuentes	Mayormente conceptos básicos (5 grupos), y errores específicos (2 grupos)
Colaboración entre pares de laboratorio	Moderada (media 3,2/5); la mínima (2 grupos), buena (3 grupos), muy buena (2 grupos)

Aspecto observado	Descripción
Metodologías clave	Mayormente ejemplos prácticos (6 grupos), seguido de explicación teórica (5 grupos) y uso de plataforma (3 grupos)
Adaptación al nivel del estudiante	Ofrece diferentes niveles (3 grupos), intenta simplificar (2 grupos), no observable (2 grupos)
Manejo de dificultades	100% reactivo (7 grupos)
Adaptación al estilo de aprendizaje del estudiante	Ofrece diferentes tipos de recursos (3 grupos), recursos visuales (2 grupos), no observable (2 grupos)
Problemas recurrentes	Lógica de programación y manejo de sintaxis (5 grupos cada uno)

La observación evidenció que las clases en laboratorio facilitan la práctica y la colaboración, pero la infraestructura limitada afecta la experiencia. La personalización de la enseñanza es escasa y la retroalimentación tiende a ser grupal.

DISCUSIÓN

a. Triangulación de los hallazgos

La combinación de entrevistas, encuestas y observaciones ofrece una visión integral del proceso de enseñanza-aprendizaje de la programación en la FCET-UAGRM. Los tres métodos coinciden en que la lógica de programación es el principal obstáculo, tanto para docentes como estudiantes. Las observaciones evidencian frustración ante problemas que requieren razonamiento lógico y una limitada disponibilidad de apoyo individualizado.

La motivación estudiantil es media, con mayor valoración de utilidad futura que inmediata, lo que sugiere una débil conexión entre contenidos y disciplina (se confirma en las bajas puntuaciones sobre “conexión con el área de estudio” y en los comentarios docentes), así como en la disminución del interés cuando no existen ejemplos contextualizados.

La demanda de personalización es clara y consistente: los estudiantes requieren materiales adaptados a su ritmo y nivel, y los docentes reconocen limitaciones para atender esta necesidad. Las altas expectativas hacia sistemas inteligentes (media 4,04) indican una actitud favorable hacia soluciones tecnológicas.

b. Comparación con estudios previos

La literatura confirma que la lógica de programación sigue siendo el principal obstáculo en niveles iniciales, en línea con estudios clásicos (Milne & Rowe, 2002), lo que sugiere un problema asociado a procesos cognitivos fundamentales más que a limitaciones tecnológicas. El pensamiento computacional destaca la necesidad de enfoques estructurados y progresivos

(Hsu y otros, 2018) y desafíos en su implementación curricular (Qian & Lehman, 2017). Además, investigación reciente evidencia la transición hacia modelos basados en datos y personalización (Rovshenov & Sarsar, 2023).

El interés por plataformas inteligentes coincide con revisiones que resaltan el potencial transformador de la IA en educación superior (Zawacki-Richter y otros, 2019), cuyas implicaciones pedagógicas han sido ampliamente discutidas (Meylani, 2024) y cuyo impacto creciente es confirmado en revisiones generales (Chen y otros, 2020). Los sistemas adaptativos han demostrado mejorar la experiencia de aprendizaje mediante retroalimentación personalizada, incluyendo evidencia de mejoras en calidad académica universitaria (Rodríguez Cairo & Ramírez Echavarría, 2023) y en entornos colaborativos (Tan y otros, 2022).

Desde la perspectiva institucional, la analítica de aprendizaje permite identificar patrones y orientar intervenciones (Romero & Ventura, 2020), aunque requiere marcos normativos adecuados (Tsai & Gasevic, 2017), siendo además estratégica en educación superior (Viberg y otros, 2018). Finalmente, la transformación digital universitaria implica integrar tecnologías emergentes con modelos pedagógicos basados en evidencia, como e-learning y m-learning, cada vez más presentes en cursos introductorios de programación (Malik y otros, 2019).

c. Limitaciones del estudio

Este diagnóstico presenta limitaciones relevantes para la interpretación de los resultados:

- La muestra fue intencional y reducida (23 estudiantes de Ingeniería Electrónica), lo que limita la generalización.
- Aunque se incluyó a los siete docentes de MAT204/INF110, el tamaño sigue siendo insuficiente para identificar tendencias pedagógicas universales.
- Los estudiantes encuestados habían aprobado la asignatura, lo que puede introducir sesgo de éxito y errores de recuerdo.
- La no participación de otros estudiantes (incluyendo quienes no aprobaron Programación I) podría implicar experiencias distintas que los voluntarios.
- El diseño transversal (solo una sesión por grupo) impide captar la variabilidad de la dinámica de aula durante el semestre.

d**.** Implicaciones para el diseño de un modelo adaptativo

Los hallazgos respaldan el diseño de un modelo adaptativo basado en IA que:

- Ofrezca ejercicios personalizados según el nivel de lógica de cada estudiante.
- Proporcione retroalimentación inmediata y específica sobre errores comunes.
- Contextualice los contenidos y ejemplos según cada ingeniería.
- Permita el seguimiento del progreso y la detección temprana de dificultades.
- Se integre o complemente a la plataforma Moodle ya utilizada por los docentes.

La alta aceptación estudiantil y la necesidad expresada por los docentes constituyen una base empírica para el diseño de dicho modelo.

CONCLUSIONES

El diagnóstico en la asignatura MAT204/INF110 de la FCET de la UAGRM permitió caracterizar el entorno educativo, la motivación y el rendimiento académico en la enseñanza de programación en ingenierías no computacionales:

- Entorno educativo: Se identificó un contexto con recursos tecnológicos limitados, metodologías predominantemente magistrales y baja personalización. Solo el 80% de los equipos de laboratorio eran funcionales y la retroalimentación fue principalmente grupal. Esto contrasta con la demanda estudiantil de personalización (media 3,57 en expectativa de ejercicios adaptados), evidenciando una brecha entre la oferta educativa y sus necesidades.

- Motivación estudiantil: La motivación es moderada (3,35/5), con mayor valoración de la utilidad futura (4,00) que de la actual (3,30), lo que refleja una débil conexión entre contenidos y carreras. Esta desconexión, también señalada por los docentes, impacta negativamente en el compromiso y esfuerzo sostenido.

- Rendimiento académico: Las principales dificultades se concentran en la lógica de programación (2,74) y en la adaptación al estilo de aprendizaje (2,65). Aunque cerca del 78% aprueba en primer intento, la nota media (63,61) es moderada-baja y casi el 22% requiere más intentos. Además, según los docentes, 50% o menos logra aplicar los conocimientos, lo que evidencia un rendimiento subóptimo.

En síntesis, los resultados confirman la necesidad de transformar la enseñanza de la programación en la FCET-UAGRM hacia un enfoque más personalizado. Esta evidencia constituye una línea base empírica para futuras investigaciones orientadas al diseño de modelos educativos adaptativos apoyados en tecnologías emergentes como la IA.

AGRADECIMIENTOS

A la Universidad Autónoma Gabriel René Moreno por el apoyo institucional, y a los docentes y estudiantes participantes por su colaboración en la recolección de datos.

A la Dra. Nelly Patricia Rojas por su dirección académica.

Al Dr. Marcos Calle por su apoyo constante a la consecución de nuestros objetivos.

REFERENCIAS BIBLIOGRÁFICAS

Almusharraf, A. (2024). An Investigation of University Students' Perceptions of Learning Management Systems: Insights for Enhancing Usability and Engagement. *Sustainability, 16*(22), e10037. https://doi.org/10.3390/su16221l0037

Bond, M., Bedenlier, S., Marín, V., & Händel , M. (2014). Emergency remote teaching in higher education: mapping the first global online semester. *International Journal of Educational Technology in Higher Education, 18*(50), 1-24. https://doi.org/10.1186/s41239-021-00282-x

Chassignol, M., Khoroshavin, A., Klimova, A., & Bilyatdinova, A. (2018). Artificial Intelligence trends in education: a narrative overview. *Procedia Computer Science, 136*(1), 16-24. https://doi.org/10.1016/j.procs.2018.08.233

Cheah, C. S. (2020). Factors Contributing to the Difficulties in Teaching and Learning of Computer Programming: A Literature Review. *Contemporary Educational Technology, 12*(2), ep272. https://doi.org/10.30935/cedtech/8247

Chen, L., Chen, P., & Lin, Z. (2020). Artificial Intelligence in Education: A Review. *IEEE Access, 8*(1), 75264-75278. https://doi.org/10.1109/ACCESS.2020.2988510

Haleem, A., Javaid, M., Qadri, M., & Suman, R. (2022). Understanding the role of digital technologies in education: A review. *Sustainable Operations and Computers, 3*(1), 275-285. https://doi.org/10.1016/j.susoc.2022.05.004

Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education, 126*(1), 296-310. https://doi.org/10.1016/j.compedu.2018.07.004

Ifenthaler, D., & Yau , J.-K. (2020). Utilising learning analytics to support study success in higher education: a systematic review. *Educational Technology Research and Development, 68*(1), 1961–1990. https://doi.org/10.1007/s11423-020-09788-z

Luxton-Reilly, A., Simon, Albluwi, I., Becker, B., Giannakos, M., Kumar, A., . . . Szabo, C. (2018). Introductory programming: a systematic literature review. *23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '18). 23*, págs. 55–106. Madrid Spain: Association for Computing Machinery (ACM). https://doi.org/10.1145/3293881.3295779

Malik, S., Mathew, R., Al-Nuaimi, R., Al-Sideiri, A., & Coldwell-Neilson, J. (2019). Learning problem solving skills: Comparison of E-learning and M-learning in an introductory programming course. *Education and Information Technologies, 24*(1), 2779–2796. https://doi.org/10.1007/s10639-019-09896-1

Meylani, R. (2024). A Critical Glance at Adaptive Learning Systems Using Artificial Intelligence: A Systematic Review and Qualitative Synthesis of Contemporary Research Literature. *Western Anatolia Journal of Educational Sciences, 15*(3), 3519-3547. https://doi.org/10.51460/baebd.1525452

Milne, I., & Rowe, G. (2002). Difficulties in Learning and Teaching Programming—Views of Students and Tutors. *Education and Information Technologies, 7*(1), 55–66. https://doi.org/10.1023/A:1015362608943

Pears, A. (2002). Defining a global view of CS education research. *7th annual conference on Innovation and technology in computer science education* (pág. 227). Aarhus Denmark: Association for Computing Machinery. https://doi.org/10.1145/544414.544495

Qian, Y., & Lehman, J. (2017). Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review. *ACM Transactions on Computing Education (TOCE), 18*(1), 1-24. https://doi.org/10.1145/3077618

Rodríguez Cairo, D., & Ramírez Echavarría, Y. (2023). Smart Tutors: improving the quality of higher education through AI. *LatIA, 1*(8), 1-10. https://doi.org/10.62486/latia20238

Romero Saldarriaga, M. A., León Galarza, L. M., & León, G. L. (2024). Impacto de la tecnología en el proceso de enseñanza aprendizaje: Un análisis integral. *Ciencia Latina Revista Científica Multidisciplinar, 8*(3), 9245-9270. https://doi.org/10.37811/cl_rcm.v8i3.12074

Romero, C., & Ventura, S. (2020). Educational data mining and learning analytics: An updated survey. *WIREs Data Mining Knowl Discovery, 10*(3), e1355. https://doi.org/10.1002/widm.1355

Rovshenov, A., & Sarsar, F. (2023). Research trends in programming education: A systematic review of the articles published between 2012-2020. *Journal of Educational Technology & Online Learning, 6*(1), 48-81. https://doi.org/10.31681/jetol.1201010

Tan, S., Lee, A., & Lee, M. (2022). A systematic review of artificial intelligence techniques for collaborative learning over the past two decades. *Computers and Education: Artificial Intelligence, 3*(1), e100097. https://doi.org/10.1016/j.caeai.2022.100097

Tsai, Y.-S., & Gasevic, D. (2017). Learning analytics in higher education --- challenges and policies: a review of eight learning analytics policies. *LAK '17: 7th International Learning Analytics and Knowledge Conference*. *17*, págs. 233-242. Vancouver British Columbia Canada: Association for Computing Machinery. https://doi.org/10.1145/3027385.3027400

Viberg, O., Hatakka, M., Bälter, O., & Mavroudi, A. (2018). The current landscape of learning analytics in higher education. *Computers in Human Behavior, 89*(1), 98-110. https://doi.org/10.1016/j.chb.2018.07.027

Watson, C., & Li, F. W. (2014). Failure rates in introductory programming revisited. *ITICSE '14: Innovation and Technology in Computer Science Education Conference* (págs. 39–44). Uppsala, Sweden: Association for Computing Machinery. https://doi.org/10.1145/2591708.2591749

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35. https://doi.org/10.1145/1118178.1118215

Zawacki-Richter, O., Marín, V., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education – where are the educators? *International Journal of Educational Technology in Higher Education, 16*(39), 1-27. https://doi.org/10.1186/s41239-019-0171-0