

UNIVERSIDAD PEDAGÓGICA NACIONAL

**MAESTRÍA TECNOLOGÍAS DE LA INFORMACIÓN APLICADAS A LA
EDUCACIÓN**

**Impacto de Prácticas Artísticas y Metacognición en la Motivación y el
Logro de Aprendizaje en Programación Frontend: un Estudio con
Estudiantes de Primer Año de Educación Universitaria**

Bogotá D.C., Colombia — 2026

UNIVERSIDAD PEDAGÓGICA NACIONAL

**MAESTRÍA TECNOLOGÍAS DE LA INFORMACIÓN APLICADAS A LA
EDUCACIÓN**

**Impacto de Prácticas Artísticas y Metacognición en la Motivación y el
Logro de Aprendizaje en Programación Frontend: un Estudio con
Estudiantes de Primer Año de Educación Universitaria**

Presentado por:
JULIAN DARIO BEJARANO GÓMEZ

Director:
LUIS SANABRIA

Bogotá D.C., Colombia — 2026

Tabla de Contenido

Tabla de Contenido.....	3
Lista de Tablas.....	5
Lista de Figuras.....	6
Introducción.....	7
Abstract.....	8
Capítulo 1. Presentación de la Investigación.....	10
1.1 Planteamiento del Problema.....	10
1.2 Pregunta de Investigación.....	12
1.3 Justificación.....	13
1.3.1 Nivel pedagógico.....	13
1.3.2 Nivel investigativo.....	14
1.3.3 Nivel práctico.....	14
1.3.4 Pertinencia curricular internacional.....	15
1.4 Objetivos de la Investigación.....	16
1.4.1 Objetivo general.....	16
1.4.2 Objetivos específicos.....	16
Capítulo 2. Antecedentes.....	18
2.1 Estrategia de búsqueda y selección de estudios.....	18
2.2 Clúster A — Prácticas artísticas y programación creativa en la enseñanza de la programación.....	20
2.3 Clúster B — Metacognición y autorregulación en el aprendizaje de programación.....	22
2.4 Clúster C — Motivación y autoeficacia en la educación en computación.....	23
2.5 Clúster D — Enfoques artísticos y marco STEAM en la educación técnica.....	24
2.6 Clúster E — Abandono y retención en cursos introductorios de programación universitaria.....	24
2.7 Magnitud de los efectos reportados en los antecedentes.....	26
2.8 Síntesis: el vacío investigativo que esta investigación ocupa.....	27
Capítulo 3. Marco Teórico.....	29
3.1 Pilar 1 — Metodologías artísticas en el aprendizaje de programación.....	29
3.1.1 Arte y código: una historia compartida.....	29
3.1.2 El construccionismo como fundamento pedagógico.....	30
3.1.3 Creative coding y herramientas de programación creativa.....	31
3.1.4 El enfoque STEAM: arte como componente estructural.....	32
3.1.5 Cognición encarnada y el aprendizaje a través de la percepción y la acción.....	32
3.2 Pilar 2 — Metacognición y autorregulación del aprendizaje.....	34

3.2.1 El concepto de metacognición: origen y definición.....	34
3.2.2 El modelo de dos niveles de Nelson y Narens.....	34
3.2.3 Autorregulación del aprendizaje: el modelo cíclico de Zimmerman.....	35
3.2.4 Metacognición en el aprendizaje de programación: especificidades del dominio	35
3.2.5 La metacognición como variable mediadora en esta investigación.....	37
3.3 Pilar 3 — Motivación en el aprendizaje de programación.....	37
3.3.1 Autoeficacia: la creencia en la capacidad personal como predictor del logro.....	37
3.3.2 Teoría de la Autodeterminación: autonomía, competencia y sentido de pertenencia.....	38
3.3.3 Teoría de orientación a metas: enfoque en maestría frente a enfoque en rendimiento.....	38
3.4 Pilar 4 — Programación Frontend en primer año universitario.....	39
3.4.1 La programación Frontend como dominio específico de aprendizaje.....	39
3.4.2 El problema estructural del abandono en cursos introductorios.....	41
3.4.3 El contexto latinoamericano.....	41
3.5 Articulación teórica: el modelo explicativo integrado.....	42
Capítulo 4. Metodología.....	43
4.1 Tipo y diseño de investigación.....	43
4.2 Variables e hipótesis.....	44
4.3 Población y muestra.....	46
4.4 Instrumentos de recolección de datos.....	48
4.4.1 MSLQ — Instrumento de medición de motivación.....	48
4.4.2 Instrumento integrado de post-test.....	49
4.4.3 Rúbricas de desempeño y portafolio de proyectos.....	50
4.5 Procedimiento por fases del diseño.....	51
4.6 Análisis de datos.....	52
4.7 Consideraciones éticas.....	53
4.8 Uso de inteligencia artificial en el proceso de investigación.....	53
4.8.1 Herramientas utilizadas.....	54
4.8.2 Criterio general de uso.....	55
4.8.3 Uso por momento de la investigación.....	55
4.8.4 Controles aplicados.....	58
4.8.5 Alcance y responsabilidad.....	59
Capítulo 5. Propuesta Pedagógica: el Curso como Intervención.....	60
5.1 Diseño instruccional: filosofía y principios.....	60
5.2 Alineación curricular con CS2023.....	61
5.3 Progresión de herramientas: de p5.js a JavaScript.....	61
5.4 El ambiente de aprendizaje en Brightspace (AVA).....	62
5.5 Estrategias metacognitivas integradas.....	70

5.6 Dimensiones de observación y seguimiento docente.....	70
Capítulo 6. Resultados y Análisis.....	72
6.1 Verificación de supuestos estadísticos.....	72
6.2 Variable de motivación: resultados MSLQ pre-post.....	73
6.3 Variable de logro de aprendizaje: resultados del instrumento integrado.....	76
6.4 Variable mediadora: metacognición.....	79
6.5 Resultados cualitativos.....	82
6.6 Prácticas artísticas: módulo de Arte Generativo (OE3).....	86
6.7 Arte Generativo y logro de aprendizaje.....	90
6.8 Prácticas artísticas: módulo de Puntillismo (OE3).....	92
Capítulo 7. Discusión y conclusiones.....	97
7.1 Discusión de resultados frente a la teoría.....	98
7.2 Verificación de hipótesis.....	104
7.3 Conclusiones por objetivo específico.....	106
7.4 Limitaciones del estudio.....	108
7.5 Contribuciones de la investigación.....	110
7.6 Recomendaciones para la implementación de la metodología.....	111
7.7 Proyecciones y líneas de trabajo futuro.....	113
Referencias Bibliográficas.....	115
Anexos.....	126
Anexo A — Instrumento MSLQ (55 ítems).....	126
Anexo B — Instrumento integrado de post-test.....	126
Anexo C — Formato de consentimientos informados.....	126
Anexo D — Encuestas de puntillismo.....	127
Anexo E — Encuestas arte generativo (previo, durante, después).....	127
Anexo F — Presentación proyecto final Arte Generativo.....	127
Anexo G — Muestra de código elaborado por estudiantes.....	127
Anexo H — Documento de uso de IA y comandos implementados.....	127

Lista de Tablas

Tabla 1. Preguntas específicas y variables de la investigación.....	13
Tabla 2. Objetivos específicos.....	16
Tabla 3. Descripción de las variables de estudio.....	44
Tabla 4. Fases de implementación.....	51
Tabla 5. Alineación curricular.....	61
Tabla 6. Comparación pre-post de las subescalas MSLQ (escala 1–7), N = 13.....	73
Tabla 7. Porcentaje de respuestas correctas por sección temática (N = 17).....	76

Tabla 8. Descriptivos de las medias por categoría de autorreporte del post-test (escala 1–5, N = 17).....	77
Tabla 9. Correlaciones de Spearman entre indicadores percibidos y logro real (N = 17)	78
Tabla 10. Correlación de Spearman entre subescalas MSLQ (post) y puntaje total del post-test integrado (N = 13).....	80
Tabla 11. Categorías emergentes en las reflexiones y entregas de estudiantes, por fase del ciclo de autorregulación (Zimmerman, 2000).....	83
Tabla 12. Descriptivos de los 10 ejes de Arte Generativo por momento (escala 1–10, N = 13).....	87
Tabla 13. Prueba de Friedman por eje (N = 13, gl = 2).....	88
Tabla 14. Comparaciones pareadas post-hoc para Q04 (N = 13).....	89
Tabla 15. Correlación de Spearman entre ejes de Arte Generativo (post) y logro real (N = 13).....	91
Tabla 16. Eje 3 — Comprensión técnica: descriptivos y prueba de Friedman (N = 10). .	93
Tabla 17. Comparaciones pareadas post-hoc para el Eje 3 — Comprensión técnica del módulo de Puntillismo (N = 10).....	93
Tabla 18. Ejes 4 y 6 del módulo de Puntillismo: prueba de Wilcoxon pre-post (N = 10)	94
Tabla 19. Descriptivos de ejes e ítems no sometidos a prueba de cambio.....	95

Lista de Figuras

Figura 1. Estructura general del Ambiente Virtual de Aprendizaje.....	64
Figura 2. Ambiente Virtual de Aprendizaje (sección inicios del arte generativo, video interactivo Vera Molnar).....	65
Figura 3. Sección de encuestas del Ambiente Virtual de Aprendizaje.....	66

Introducción

Aprender a programar, para un estudiante de primer año, rara vez es solo un ejercicio técnico: es también el momento en que muchos deciden, sin decirlo en voz alta, si esto es "lo suyo" o no. Esta investigación surge de una pregunta cotidiana en el espacio de enseñanza: ¿qué pasa si, en lugar de enseñar código de forma aislada, se enseña a través del arte, con pausas que le permitan al estudiante pensar en cómo se aprende?

El estudio se realizó con estudiantes de primer año del programa Creación Digital de la Universidad El Bosque (Bogotá, Colombia), bajo un diseño pre-experimental pre-post de un solo grupo, conformado por un grupo natural de aula. A lo largo de un semestre los estudiantes recorrieron cuatro módulos artísticos: pensamiento algorítmico, puntillismo, abstraccionismo y arte generativo. Cada módulo funcionó como puerta de entrada a JavaScript, HTML y CSS e incorporó preguntas metacognitivas en el propio flujo de trabajo creativo. La motivación y la autoeficacia se midieron con el MSLQ (N = 13 pares completos); el logro técnico, con un instrumento integrado de post-test (N = 17).

La motivación intrínseca y la autoeficacia no cambiaron de manera significativa; la autorregulación metacognitiva contra toda expectativa descendió. Una posible forma de asumir este resultado es que, luego de enfrentar la dificultad real de programar, los estudiantes se autoevalúan con más exigencia y no con más confianza, lo cual puede ser una característica al inicio del curso. El desempeño técnico general obtuvo resultados óptimos (81.5%), encontrando que quienes registraron un mayor avance en su propio aprendizaje

fueron también quienes mejores resultados obtuvieron en sus pruebas técnicas. Esta tesis documenta esa complejidad con transparencia y aporta a un diálogo entre la disciplina artística, la metacognición y el aprendizaje de programación en el contexto universitario latinoamericano.

El documento se organiza en siete capítulos; el primero presenta el problema, la pregunta y los objetivos de la investigación; el segundo revisa los antecedentes empíricos en cinco ejes o clústeres temáticos; en el tercer capítulo se desarrolla el marco teórico en cuatro pilares; el cuarto capítulo expone el diseño metodológico y los instrumentos implementados; el quinto describe la propuesta pedagógica desarrollada en el aula; el sexto presenta los resultados cuantitativos y cualitativos; y finalmente en el séptimo capítulo se presenta la discusión, los hallazgos, adicionalmente, se presenta el resultado de verificación de las hipótesis y se plantean las conclusiones, limitaciones y proyecciones del estudio.

Palabras clave: educación artística, programación creativa, metacognición, motivación intrínseca, autoeficacia, Frontend, creative coding, STEAM, educación universitaria.

Abstract

For a first-year university student, learning to program is rarely a purely technical exercise. It is also the moment when many quietly decide whether coding is “for them.” This study grew out of that everyday classroom question: what happens when code is taught through art rather than in isolation, with explicit pauses to think about how one is learning?

The study took place with first-year students of the Digital Creation program at Universidad El Bosque (Bogotá, Colombia), under a single-group pre-experimental pre-post design with an intact classroom group. Over one semester, students worked through four artistic modules: algorithmic thinking, pointillism, abstract art, and generative art. Each module served as a gateway to JavaScript, HTML, and CSS, and embedded metacognitive prompts in the creative workflow. Motivation and self-efficacy were measured with the MSLQ (N = 13 complete pairs); technical achievement, with an integrated post-test instrument (N = 17). The results do not tell a simple story of success, and that is precisely where their honesty lies. Intrinsic motivation and self-efficacy showed no measurable change. Metacognitive self-regulation, against all expectations, declined; one plausible reading is that, after facing the real difficulty of programming, students came to judge themselves more demanding rather than more confidently. Overall technical performance was nevertheless solid (81.5%), and the students who perceived the greatest progress in their own learning were indeed those who performed best. This thesis documents that complexity transparently, as a situated contribution to the dialogue between art, metacognition, and programming education in the Latin American university context.

Keywords: creative coding, metacognition, intrinsic motivation, self-efficacy, Frontend programming, STEAM, undergraduate education, first-year students.

Capítulo 1. Presentación de la Investigación

Esta investigación se inscribe en la enseñanza de la programación en el ámbito universitario y la educación mediada por las artes. Su propósito es analizar el impacto de una metodología pedagógica que integra prácticas artísticas y procesos metacognitivos sobre la motivación autopercebida y el logro de aprendizaje de estudiantes de primer año en programación Frontend de nivel básico. Los apartados que siguen exponen los fundamentos, el diseño y los instrumentos del estudio.

1.1 Planteamiento del Problema

La enseñanza de programación en el nivel universitario enfrenta un problema estructural en los niveles introductorios, según las revisiones encontradas se sitúan entre el 30% y el 50% la proporción de estudiantes que no completan satisfactoriamente su primer curso de programación (Bennedsen & Caspersen, 2007; Robins et al., 2003). Al problema de permanencia se suma uno de orden motivacional; la autoeficacia inicial, relacionada como la creencia en la propia capacidad para aprender y resolver problemas de programación suele ser baja entre quienes llegan sin experiencia previa y puede ser este factor un determinante o predictor de la decisión del estudiante de continuar o abandonar el curso (Lishinski et al., 2016). Tomić et al. (2025) documentaron además la ansiedad ante la programación como un constructo propio del dominio en una muestra de 541 estudiantes de primer año.

El problema se hace mayor donde la programación se enseña de forma puramente técnica y resulta alejada de experiencias significativas de los aprendices. Martins et al. (2018) identificaron la baja motivación y las dificultades de pensamiento abstracto como las dos causas principales del fracaso en cursos universitarios brasileños de programación. Algunas razones propuestas sugieren que sin anclaje en contextos que el estudiante reconozca como propios o cercanos, la relación con el código se da desde una distancia cognitiva y también emocional respecto de aquello que pretende aprender.

La integración de metodologías artísticas en el aprendizaje de la programación se ha explorado desde el creative coding y desde el enfoque construccionista (Papert, 1980; Resnick, 2007), con resultados prometedores en compromiso y motivación. En esa misma línea, Greenberg et al. (2012) observaron mayor persistencia ante el error entre estudiantes de CS1 que trabajaban con portafolios visuales creativos; Jawad et al. (2018) registraron mejoras significativas de logro en un diseño con grupo control que integró arte y animación; y Graßl et al. (2025) documentaron cambios en las creencias sobre el perfil de quien se dedica a programar tras un curso basado en arte generativo. Ahora bien, sigue faltando evidencia sistemática sobre el impacto de esta integración en variables medibles de motivación y logro. La carencia se hace más grande cuando se habla de programación Frontend y en el contexto universitario latinoamericano, donde la producción disponible es escasa (Pereira et al., 2020; Ramírez-Echeverry et al., 2025).

Paralelamente, la metacognición entendida como el conjunto de procesos mediante los cuales el aprendiz monitorea, regula y evalúa su propio pensamiento (Flavell, 1976), ha indicado ser un predictor confiable del desempeño académico, con correlaciones con el logro que se sostienen más allá de la inteligencia general (Veenman et al., 2006). En el dominio específico de la programación, Loksa et al. (2016) mostraron experimentalmente que los estudiantes que reciben una orientación metacognitiva alcanzan mejor desempeño y mayor autoeficacia que quienes resuelven las mismas tareas sin ese andamiaje. De este panorama surge el enfoque del presente estudio y corresponde a indagar si puede la combinación de prácticas artísticas y estrategias metacognitivas constituir una metodología que permita mejorar la motivación y el logro de aprendizaje en programación.

1.2 Pregunta de Investigación

La presente investigación se orienta en torno a la siguiente pregunta principal:

¿Cuál es el impacto de una metodología que integra prácticas artísticas y procesos metacognitivos sobre la motivación y el logro de aprendizaje en programación Frontend de estudiantes de primer año universitario?

De esta pregunta central se subdividen cuatro preguntas específicas que corresponden a las variables y dimensiones del diseño:

Tabla 1*Preguntas específicas y variables de la investigación*

Pregunta específica	Variable que aborda
¿Qué cambios se observan en la percepción que los estudiantes tienen de su motivación intrínseca y de su autoeficacia tras la implementación de la metodología?	Motivación (autoeficacia + orientación intrínseca/extrínseca)
¿En qué medida la metodología se asocia con el logro de aprendizaje en programación Frontend, medido a través del instrumento integrado de post-test?	Logro de aprendizaje en Frontend
¿De qué manera las prácticas artísticas se relacionan con la activación de procesos metacognitivos autopercebidos que median entre la metodología y los resultados?	Metacognición como variable mediadora
¿Cómo perciben los estudiantes la experiencia de aprender programación a través de un enfoque artístico y metacognitivo?	Experiencia subjetiva / dato cualitativo complementario

1.3 Justificación

1.3.1 Nivel pedagógico

Existe una necesidad documentada en el contexto de la enseñanza de programación en niveles iniciales y corresponde a la renovación de las metodologías para el aprendizaje. Las aproximaciones puramente técnicas no logran sostener la motivación ni garantizar comprensión profunda por parte de los estudiantes, es por esta razón que explorar metodologías que integren el arte como vehículo de aprendizaje y la metacognición como andamiaje reflexivo representa una alternativa pedagógicamente fundamentada y coherente con los principios del aprendizaje significativo, el construccionismo y la Teoría de la Autodeterminación (Ryan & Deci, 2000).

1.3.2 Nivel investigativo

La presente investigación aporta conocimiento sobre la intersección entre tres campos: metodologías artísticas, metacognición y aprendizaje de programación. Esta posible articulación en estos campos ha sido poco estudiada de forma sistemática. Los resultados contribuyen a la literatura sobre pedagogía del código, didáctica universitaria y diseño instruccional creativo, con especial relevancia para el contexto latinoamericano donde la producción investigativa en este dominio es escasa y se hace necesario el conocimiento de diversas aproximaciones en estos campos.

1.3.3 Nivel práctico

Los hallazgos tienen la posibilidad de dar claridades sobre la aplicación directa en las propuestas de diseños de cursos de programación para estudiantes de primer año en el contexto universitario quienes llegan con pocas o nulas bases prácticas y conceptuales en el campo de la programación. Los resultados de la intervención ofrecen aproximaciones didácticas y posibles elementos teóricos para la construcción de un modelo replicable y adaptable para docentes de áreas STEAM que permitan mejorar la experiencia y los resultados de sus estudiantes en cursos de programación en niveles introductorios.

1.3.4 Pertinencia curricular internacional

Un elemento adicional que refuerza la justificación de la presente investigación es la coherencia de la metodología artística implementada con estándares curriculares internacionales vigentes como lo es el currículo guía CS2023, el cual fue producido en conjunto por la ACM, el IEEE Computer Society y la AAI y ratificado en 2024, en esta guía se organiza el cuerpo de conocimiento de la computación en 17 Knowledge Areas (KA) distribuidas en tres áreas de competencia (competency areas): Software Development, Systems Development y Applications Development (ACM/IEEE-CS/AAAI Joint Task Force, 2024). El análisis de alineación entre los contenidos del curso y dicho marco curricular permite observar que los cuatro módulos temáticos propuestos en el curso de la investigación: pensamiento algorítmico (Rosas danst Rosas), puntillismo y el pixel digital, abstraccionismo y pioneros del arte generativo, tienen incidencia en 9 de las 17 KA del CS2023, incluyendo cuatro que pertenecen al CS Core obligatorio para toda persona que se haya graduado en áreas relativas a la computación: Algorithmic Foundations (AL), Software Development Fundamentals (SDF), Mathematical and Statistical Foundations (MSF) y Artificial Intelligence (AI). Esta amplitud de posibles campos de vínculo y acción permite cuestionar el argumento de que las metodologías artísticas en programación sacrifican rigor técnico en favor de motivación, siendo quizás un argumento que deja de lado la posibilidad que brindan las prácticas artísticas para un vínculo profundo con procedimientos técnicos, sensibles, prácticos y conceptuales.

1.4 Objetivos de la Investigación

1.4.1 Objetivo general

Determinar el impacto de una metodología que integra prácticas artísticas y procesos metacognitivos sobre la motivación autopercebida y el logro de aprendizaje en programación Frontend básica de estudiantes de primer año de educación universitaria, mediante un diseño pre-experimental pre-post con un solo grupo.

1.4.2 Objetivos específicos

Tabla 2
Objetivos específicos

Nº	Objetivo específico	Variable asociada
OE1	Caracterizar la percepción que los estudiantes tienen de su propia motivación (autoeficacia y orientación a metas intrínsecas y extrínsecas) antes y después de la intervención, a partir de sus autorreportes en el MSLQ.	Motivación autopercebida y medición pre y post
OE2	Evaluar el logro de aprendizaje en programación Frontend mediante la progresión instruccional p5.js → JavaScript puro (variables, funciones, estructuras de control, DOM, eventos), usando pruebas técnicas, proyectos entregables, un instrumento integrado de post-test alineado con los contenidos de nivel CS Core del currículo CS2023 (SDF, AL, GIT).	Logro de aprendizaje e instrumento cuantitativo
OE3	Describir las prácticas artísticas implementadas y su relación con el desarrollo de habilidades de programación Frontend en el aula.	Metodología artística y documentación del proceso
OE4	Analizar la relación entre la autorregulación metacognitiva autopercebida por los estudiantes y los cambios observados en su motivación declarada y en su logro de aprendizaje.	Metacognición autopercebida (variable mediadora)
OE5	Documentar las percepciones de los estudiantes sobre su propio proceso de aprendizaje, a partir del análisis de contenido de las bitácoras producidas en las actividades	Dato cualitativo — registro del estudiante

Tabla 2
Objetivos específicos

artísticas del curso.

Capítulo 2. Antecedentes

Cinco clústeres o ejes temáticos organizan la revisión de literatura que sustenta esta investigación. Se construyeron en correspondencia con las variables y las dimensiones del modelo explicativo propuesto en el marco teórico, y abordan, respectivamente: (A) el uso de prácticas artísticas y programación creativa en la enseñanza de la programación; (B) la metacognición y la autorregulación en contextos de aprendizaje de código; (C) la motivación y la autoeficacia en la educación en computación; (D) los enfoques artísticos y el marco STEAM en la educación técnica; y (E) el abandono y la retención en cursos introductorios de programación universitaria. La búsqueda se realizó en Scopus, ERIC, ACM Digital Library, Semantic Scholar, Scholar Gateway y Redalyc, dentro de una franja temporal entre 2012 y 2026, y dio prioridad a estudios empíricos revisados por pares.

2.1 Estrategia de búsqueda y selección de estudios

Este capítulo se apoya en una revisión temática de la literatura organizada por conglomerados y no en una revisión sistemática con protocolo PRISMA. La distinción no es un tecnicismo, puesto que delimita lo que puede afirmarse a partir del corpus reunido: el propósito fue mapear el estado del campo en torno a las variables del estudio, no estimar un efecto agregado.

Se consultaron seis bases de datos: Scopus, ERIC, ACM Digital Library, Semantic Scholar, Scholar Gateway y Redalyc. Se estableció un periodo de búsqueda entre el 2012 y 2026,

dando prioridad en estudios empíricos revisados por pares. Redalyc se incluyó para recuperar producción latinoamericana, que es uno de los vacíos que en este estudio se identifica.

Los criterios de inclusión aplicados fueron tres: estudios empíricos con datos primarios o revisiones sistemáticas publicados entre 2012 y 2026; trabajos centrados en la enseñanza o el aprendizaje de la programación, en la integración de prácticas artísticas o del enfoque STEAM, en la metacognición y la motivación en educación en computación y publicaciones en inglés o español. Se excluyeron, en cambio, los trabajos exclusivamente conceptuales cuando existía evidencia empírica disponible sobre el mismo asunto, los estudios sobre integración artística sin componente de programación o pensamiento computacional y los documentos sin revisión por pares.

A ello se sumó el rastreo de referencias citadas, o bola de nieve, a partir de los trabajos seminales de cada categoría. Ese procedimiento permitió incorporar antecedentes al estudio que pudiesen relacionarse con los temas y que fuesen anteriores a la franja temporal cuando su pertinencia y relación lo justificaban.

El procedimiento produjo un corpus de 37 documentos recuperados y revisados a texto completo. De ellos se citan 23 en este capítulo. Los 14 restantes se descartaron por tener marcos conceptuales generales sin evidencia empírica sobre el objeto específico, o bien a estudios de integración artística sin componente de programación. Por conglomerado, la distribución fue la siguiente: prácticas artísticas y programación creativa (11 documentos),

metacognición y autorregulación en programación (6), motivación y autoeficacia en educación en computación (4), enfoques artísticos y marco STEAM en educación técnica (11) y abandono y retención en cursos introductorios (4). Un documento se ubica simultáneamente en los conglomerados de metacognición y motivación.

Conviene declarar una limitación del procedimiento. No se llevó un registro sistemático del número de resultados arrojados por cada ecuación en cada base de datos, de modo que la revisión no es reproducible en sentido estricto. La selección se realizó teniendo en cuenta la pertinencia temática respecto de las variables del estudio, con criterio del investigador y sin un protocolo de tamizaje por pares. Esta limitación acota, en consecuencia, el alcance de cualquier afirmación sobre la exhaustividad de búsqueda de la literatura revisada.

2.2 Clúster A — Prácticas artísticas y programación creativa en la enseñanza de la programación.

La integración de prácticas artísticas en la enseñanza de la programación tiene sus antecedentes más relevantes a comienzos de la década de 2000 con el desarrollo de Processing y p5.js y que luego daría origen al movimiento del creative coding, consolidado académicamente como entornos que permiten usar el código como medio de expresión visual e interactiva. Greenberg, Kumar y Xu (2012) implementaron portafolios visuales creativos en cursos CS1 y documentaron que los estudiantes que aprendían programación mediante la creación de proyectos que iban mas allá del dominio técnico sino que también

predominaba una intención estética, en esta propuesta los estudiantes mostraban mayor persistencia ante los errores y mayor satisfacción con el proceso de aprendizaje. Fragapane y Standl (2021) sistematizaron el campo con una propuesta de marco teórico para el creative coding en la educación en computación, argumentando que este enfoque no constituye simplemente una estrategia motivacional adicional sino una transformación epistémica del rol del código en el aula: de herramienta funcional a lenguaje expresivo.

Graßl, Musliu y Fraser (2025) estudiaron el efecto de la programación artística sobre los estereotipos asociados a quien programa y sobre las actitudes afectivas de estudiantes de secundaria. Su hallazgo principal es que la exposición a proyectos de programación con intención artística modifica de forma significativa las creencias previas acerca de quién puede programar. Bares, Manaris, McCauley y Moore (2019) documentaron, por su parte, que los enfoques creativos en computación mejoran el equilibrio de género y las tasas de retención, en especial entre poblaciones subrepresentadas. Un tercer estudio, a cargo de Jawad, Tout, Abualkibash y Xie (2018), encontró mejoras significativas de logro en el grupo experimental de cursos de secundaria que integraron arte y animación en la enseñanza de programación.

Knochel y Patton (2015) articularon la relación entre la educación artística crítica, el pensamiento computacional y el código creativo, argumentando que las prácticas artísticas activan procesos cognitivos que no necesariamente están dirigidos hacia una instrucción puramente técnica. Kafai y Burke (2014) fundamentaron teóricamente este argumento desde

el construccionismo. Los estudios de este eje temático confirman que la integración de prácticas artísticas en la enseñanza de la programación tiene efectos positivos documentados sobre la motivación, la persistencia y el logro de aprendizaje, aunque la mayoría se concentra en educación K-12 o en contextos anglosajones.

2.3 Clúster B — Metacognición y autorregulación en el aprendizaje de programación

El trabajo más relevante de este eje temático es el de Loksa, Ko, Jernigan, Oleson, Mendez y Burnett (2016), quienes demostraron experimentalmente que los estudiantes con orientación metacognitiva explícita durante la resolución de problemas de programación alcanzaban mayor desempeño y mayor autoeficacia que quienes resolvían los mismos problemas sin ese andamiaje. Prather, Pettit, Becker, Denny, Loksa y colaboradores (2019) examinaron después qué ocurre cuando el andamiaje metacognitivo se ofrece en el momento de leer el enunciado del problema: los estudiantes cometían significativamente menos errores conceptuales. Loksa, Margulieux, Becker, Craig, Denny, Pettit y Prather (2022) documentaron esta línea en una revisión sistemática, cuya conclusión resulta directamente aplicable al presente estudio: las intervenciones metacognitivas más efectivas son las que se integran en el flujo de trabajo del estudiante.

Pechorina, Anderson y Denny (2023) desarrollaron Metacodenition el cual es un entorno de programación que integra andamiaje metacognitivo estructurado, basados en este ambiente

encontraron mejoras significativas en el desempeño de los estudiantes. En un diseño pre-post con 269 estudiantes, Smit, Schmid y Robin (2024), encontraron que el disfrute durante tareas creativas de programación visual predice aumentos en la autoeficacia para programar. La revisión de este eje temático revela que la metacognición en programación ha sido estudiada principalmente en contextos de resolución de problemas algorítmicos, dejando de lado o sin una exploración a profundidad el rol de las prácticas artísticas como metodologías activadoras de procesos metacognitivos en estudiantes con poca o nula experiencia.

2.4 Clúster C — Motivación y autoeficacia en la educación en computación

En un estudio de Lishinski, Yadav, Good y Enbody (2016) se documentó que la autoeficacia inicial y la orientación a metas pueden predecir con mayor veracidad el desempeño final que la experiencia previa en computación, adicionalmente, la trayectoria de la autoeficacia a lo largo del semestre es el predictor más potente de la decisión de continuar o abandonar la carrera. Katz, Eilat y Nevo (2013) exploraron la relación entre motivación intrínseca y persistencia en tareas desafiantes. Adicionalmente, Cheryan, Master y Meltzoff (2015) documentaron que los estereotipos culturales sobre quién puede programar actúan como posibles limitantes de la motivación intrínseca. Cheng, Poon, Lau y Zhou (2019) establecieron correlaciones significativas entre el uso de estrategias de autorregulación y el logro académico en programación universitaria. La evidencia de la presente categoría converge en que la motivación en CS introductorio es un sistema dinámico que responde a las condiciones pedagógicas del entorno de aprendizaje.

2.5 Clúster D — Enfoques artísticos y marco STEAM en la educación técnica

Aguilera y Ortiz-Revilla (2021) realizaron una revisión sistemática sobre STEM versus STEAM y creatividad estudiantil, concluyendo que la integración artística produce efectos positivos consistentes sobre el pensamiento creativo y la motivación, aunque también se señala que muchos estudios carecen de mediciones rigurosas de esos resultados. Conradt, Sotiriou y Bogner (2020) documentaron que los módulos STEAM que integran componentes creativos intervienen positivamente en la autoeficacia y la motivación. Corrigan, Wong, Grove, Andersen y Hughes (2025) aportaron evidencia experimental que indica que el enfoque STEAM es el que tiene mayor efectividad para producir una comprensión conceptual de manera mas profunda.

Bautista y colaboradores (2025) introdujeron el concepto del 'artless-STEAM effect': la integración superficial del arte como decoración sin aporte cognitivo real, distinción relevante para la presente investigación cuya intervención usa el arte como andamiaje conceptual genuino y relevante mas alla de un único propósito superficial.

2.6 Clúster E — Abandono y retención en cursos introductorios de programación universitaria

Desde el contexto brasileño, Martins, Concilio y Guimarães (2018) reportaron tasas altas de fracaso y evasión en cursos de programación universitaria, las causas que se identificaron por parte de estos autores corresponden a la baja motivación y dificultades de pensamiento

abstracto en los estudiantes. López-García, Urquiza-Fuentes, Mendes y Caeiro-Rodríguez (2025) confirmaron el problema en un contexto hispanohablante con 405 estudiantes de España y Portugal. Por otra parte, Pereira, Oliveira y colaboradores (2020) analizaron datos de 2058 estudiantes en la Universidad Federal del Amazonas durante tres años, los resultados de este estudio aportan la evidencia empírica en el contexto latinoamericano de mayor escala disponible hasta el momento. Tomić, Stojanović, Antović y Milić (2025) estudiaron la ansiedad ante la programación en 541 estudiantes de primer año, definiendo la programming anxiety como un constructo específico directamente pertinente para la hipótesis de reducción de ansiedad de esta investigación. Malik y colaboradores (2021) documentaron que una intervención pedagógica estructurada redujo la tasa de abandono del 21% al 11.1%.

El antecedente de mayor relevancia contextual es el de Ramírez-Echeverry, Restrepo-Calle y Torres Jiménez (2025): investigadores colombianos que aplicaron el CEAPC, el cual es un cuestionario análogo al MSLQ, en estudiantes universitarios de programación en Colombia, encontrando que la metacognición es interdependiente con la práctica y la resolución de problemas y que los estudiantes de cursos introductorios necesitan apoyo específico en estrategias organizacionales. Este antecedente comparte contexto nacional, dominio e instrumento conceptual con la presente investigación.

2.7 Magnitud de los efectos reportados en los antecedentes

La evaluación de la magnitud de los efectos reportados en los antecedentes es pertinente para este estudio, que da un peso explícito al tamaño del efecto en el análisis de sus propios resultados. El examen del corpus arroja un panorama desigual.

En el conglomerado de metacognición, Zhang et al. (2025) reportan el efecto de mayor magnitud del corpus: la combinación de aprendizaje basado en problemas con programación en pareja elevó la habilidad metacognitiva frente al grupo control ($M = 3.52$, $DE = 0.19$ y $M = 3.19$, $DE = 0.14$, respectivamente), con $F = 86.85$, $p < .001$ y η^2 parcial = .497, es decir, cerca de la mitad de la varianza explicada por la condición experimental.

En el conglomerado STEAM, Corrigan et al. (2025) reportan efectos entre moderados y grandes sobre la comprensión conceptual: η^2 parcial = .18 y .16 para los cambios según la alineación conceptual inicial de los estudiantes, y η^2 parcial = .20 para el incremento total en la comprensión canónica; el efecto del método instruccional en el contraste multivariado fue menor (η^2 parcial = .08). Conradt y Bogner (2019), con 125 estudiantes y un diseño pre-post-retención, documentaron una ganancia de conocimiento significativa y sostenida (Wilcoxon, $Z = 7.26$, $p < .001$ entre la medición inicial y la posterior; $Z = 3.48$, $p < .001$ en la retención), mientras que las subescalas de creatividad, tras subir inicialmente, regresaron al nivel de partida ($Z = 2.74$, $p = .006$ y $Z = 2.63$, $p = .009$). Ese contraste entre una ganancia cognitiva duradera y un autorreporte disposicional que no presenta significancia resulta pertinente para la interpretación de los resultados de esta investigación (sección 7.1).

En el conglomerado de abandono y retención, Malik et al. (2021) expresan la magnitud de su intervención en términos de tasa y no de índice estandarizado: la deserción descendió del 21 % al 11.1 %.

Debe señalarse que buena parte del corpus no informa los tamaños del efecto y se limita a reportar la significación estadística, lo que impide la posibilidad de comparar magnitudes entre los estudios. La observación coincide con la de Aguilera y Ortiz-Revilla (2021), quienes en su revisión sistemática sobre STEAM advierten que numerosos trabajos del campo carecen de mediciones rigurosas en los resultados. Ese vacío refuerza una decisión metodológica de esta investigación: reportar el tamaño del efecto en todos los análisis, con independencia del valor p.

2.8 Síntesis: el vacío investigativo que esta investigación ocupa

La revisión de los cinco clústeres permite identificar tres vacíos sistemáticos que definen el lugar en que se inscribe de esta investigación. El primer vacío que se identifica está relacionado con la articulación, debido a que ningún estudio disponible ha abordado simultáneamente las prácticas artísticas, la metacognición y la motivación como un sistema integrado de variables en el contexto de la programación introductoria universitaria. El segundo vacío identificado es de dominio, ya que la programación Frontend no ha sido objeto de estudio específico en la literatura sobre creative coding, STEAM o intervenciones metacognitivas en programación en los niveles de cursos introductorios. En tercer lugar, se identifica un vacío de orden contextual, pues la investigación en aprendizaje de

programación mediada por metodologías artísticas en contextos latinoamericanos es escasa.

Por lo tanto, esta investigación se ubica en la posibilidad de explorar estos tres vacíos.

Capítulo 3. Marco Teórico

La investigación se sustenta en cuatro pilares teóricos que se articulan alrededor de la curiosidad de analizar y explorar cómo es el proceso de aprendizaje cuando está mediado por metodologías artísticas y la reflexión metacognitiva, de manera particular en este estudio, el aprendizaje en el área de programación .

3.1 Pilar 1 — Metodologías artísticas en el aprendizaje de programación

3.1.1 Arte y código: una historia compartida

La relación entre arte y programación tiene raíces profundas que van más allá de una posible casualidad. Por mencionar algunos ejemplos, en los primeros experimentos de arte computacional de la década de 1960, figuras fundamentales para el arte generativo y el desarrollo de gráficas por computador, como Vera Molnár, Georg Nees y Frieder Nake, emplearon plotters y algoritmos matemáticos para generar composiciones visuales, sentando con esto las bases para dotar al código de una dimensión técnica y como lenguaje, con todo su potencial expresivo y creativo (Galanter, 2003). El arte generativo primero, y el creative coding después, permitieron conceptualizar la programación como una práctica capaz de producir resultados creativos. Estos son solo dos ejemplos dentro de muchos en que el arte ha operado en diálogo cercano con las tecnologías emergentes de su época y en esa relación ha posibilitado nuevas rutas de exploración artística, científica y tecnológica. Algunas características del enfoque experimental del arte, adquieren relevancia particular en el

ámbito educativo, porque cambian el estatuto del error y su sesgo negativo. Al programar, un error no implica necesariamente un resultado descartable: modificar un valor puede alterar la posición, el tamaño o la repetición de una figura y generar una composición visualmente atractiva. El estudiante frente al error, asumiendo una perspectiva creativa y experimental; observa, analiza, reflexiona y decide entonces si conserva el resultado obtenido, o lo modifica o si quizás, vuelve a experimentar a partir de lo percibido en alguna parte del proceso. El error deja así de ser únicamente algo que corregir y pasa a formar parte de un sistema dinámico donde la exploración, reflexión y posibilidad de modificar y crear están presentes.

3.1.2 El construccionismo como fundamento pedagógico

Para comprender la integración del arte en el aprendizaje de la programación, el construccionismo de Seymour Papert (1980) es una referencia de alto valor ya que se propone que el aprendizaje implica no solo la adquisición de conceptos, sino también la creación de objetos que pueden ser observados, compartidos, discutidos y transformados. Esta aproximación resulta especialmente pertinente para esta investigación, donde los estudiantes no adquieren conocimientos de HTML, CSS y JavaScript a través de ejercicios aislados, sino mediante la elaboración de composiciones visuales. Asimismo, logran animaciones y piezas interactivas haciendo uso de la librería p5.js para luego pasar a la integración con el DOM y la interactividad. De esta manera, el código revela su potencia como medio y herramienta con la cual se logran materializar decisiones e intenciones

artísticas y estéticas. Durante este trayecto, la programación trasciende su dimensión técnica y se integra en una práctica creativa que permite al estudiante experimentar, modificar sus artefactos y tomar decisiones sobre los resultados, con un componente adicional: la posibilidad de sistematización.

3.1.3 Creative coding y herramientas de programación creativa

El creative coding hace referencia a un conjunto de prácticas, herramientas y comunidades centradas en el uso del código como medio de expresión y creación. Processing ha sido durante años una herramienta de referencia para artistas que llevan el código a sus indagaciones y apuestas creativas. Reas y Fry (2007) lo desarrollaron para que diseñadores y artistas pudieran programar sin lidiar con la infraestructura habitual de los lenguajes de programación y algunos fines predeterminados. En 2014, McCarthy creó p5.js como una reinterpretación de Processing para que su ejecución fuese posible en el entorno del navegador web. Esta biblioteca permite escribir código con retroalimentación visual inmediata pasando por alto la gestión de la estructura completa de un documento HTML. En el diseño instruccional de la presente intervención, p5.js además de ser la herramienta inicial de exploración, opera como un entorno de andamiaje, en consonancia con el principio de reducción de la carga cognitiva extrínseca (Sweller, 1988), la transición hacia JavaScript puro y la integración DOM se realiza de manera progresiva. La evidencia empírica disponible explorada no examina p5.js en particular, sin embargo, en los casos reportados se implementan entornos de programación visual con retroalimentación inmediata. Greenberg,

Kumar y Xu (2012) documentaron mayor persistencia ante el error y mayor satisfacción con el proceso entre estudiantes de CS1 que trabajaron con portafolios visuales creativos. Smit, Schmid y Robin (2024) encontraron además que el disfrute experimentado durante tareas de programación visual predice incrementos en la autoeficacia para programar.

3.1.4 El enfoque STEAM: arte como componente estructural

La integración de las artes en la educación STEM (transición de STEM a STEAM) ha adquirido una presencia relevante y un incremento en la literatura educativa (Henriksen, 2014; Perignat & Katz-Buonincontro, 2019). El argumento principal sostiene que el arte no debe incorporarse como un elemento decorativo o una motivación superficial, mas aún los procesos cognitivos y creativos característicos de las prácticas artísticas, tales como la observación detallada, la experimentación formal, la tolerancia a la ambigüedad y la revisión iterativa, constituyen también procesos fundamentales en el pensamiento científico y computacional.

3.1.5 Cognición encarnada y el aprendizaje a través de la percepción y la acción

La teoría de la cognición encarnada (Varela, Thompson & Rosch, 1991) sostiene que el conocimiento no consiste en un conjunto de representaciones abstractas almacenadas en la mente sino que surge de la interacción dinámica entre el individuo, su cuerpo y el entorno. La relevancia de este marco para el presente estudio no está en la tesis general, que por sí sola aporta poco a la enseñanza de la programación, sino en una consecuencia específica: lo

que Papert (1980) denominó sintonía corporal, es decir, la idea de que un concepto se comprende mejor cuando el estudiante puede situarse en el lugar del objeto manipulado y razonar desde esa perspectiva. Barsalou (2008) reunió evidencia a favor de esa idea bajo la noción de cognición fundamentada, según la cual los conceptos abstractos se apoyan en simulaciones de estados perceptuales y motores. Wilson (2002) advirtió, sin embargo, que bajo el rótulo de cognición encarnada conviven afirmaciones de una solidez muy desigual y que solo algunas cuentan con respaldo empírico robusto. Este estudio se apoya en la versión más reservada y mejor respaldada de las que identifica Wilson: aquella según la cual la percepción y la acción estructuran la comprensión conceptual.

Esa versión se traduce directamente al diseño de las actividades de la implementación, ya que el estudiante antes de programar el bucle que genera una trama puntillista, observa la obra de referencia, analiza y describe los posibles factores que modifican el tamaño y la posición de los puntos, su distribución en el lienzo y los colores predominantes. La estructura iterativa se reconoce primero en la percepción de la imagen y después se va traduciendo en la sintaxis del código. Sung, Ahn y Black (2017) documentaron un efecto semejante al introducir el pensamiento computacional mediante actividades de encarnación, en este estudio consiguieron mejoras en la comprensión de conceptos abordados antes de forma simbólica. La contribución de este marco, en suma, no consiste en afirmar que el cuerpo participa en la cognición, sino en fundamentar por qué en cada módulo del curso el

contacto con el referente artístico, las percepciones y sensaciones de este encuentro, se anticipan al proceso de codificación.

3.2 Pilar 2 — Metacognición y autorregulación del aprendizaje

3.2.1 El concepto de metacognición: origen y definición

El término metacognición fue adoptado por Flavell en 1976 para referirse al conocimiento que posee un individuo sobre sus propios procesos cognitivos y también, sobre aquellos productos que resultan de dichos procesos. Brown (1987) sistematizó la diferencia entre el conocimiento metacognitivo (declarativo, procedimental y condicional) y la regulación metacognitiva (planificación, monitoreo y evaluación). Esta estructura posiciona a la metacognición como un constructo de gran relevancia educativa ya que no basta con que el estudiante reconozca su falta de comprensión; también debe ser capaz de intervenir activamente para mejorarla.

3.2.2 El modelo de dos niveles de Nelson y Narens

Nelson y Narens (1990) propusieron un modelo de arquitectura cognitiva compuesto por dos niveles: el nivel-objeto, en el que se desarrollan los procesos cognitivos en curso, y el meta-nivel en el que el individuo mantiene un modelo dinámico de lo que ocurre en el nivel-objeto y mantiene control sobre dichos procesos. El monitoreo consiste en la transferencia de información del nivel-objeto al meta-nivel, mientras que el control implica la dirección inversa. La eficacia del aprendizaje depende de la fluidez de este ciclo en ambas direcciones.

3.2.3 Autorregulación del aprendizaje: el modelo cíclico de Zimmerman

El modelo cíclico de Zimmerman (2000) conceptualiza el aprendizaje autorregulado como un proceso que se desarrolla en tres fases iterativas: previsión, ejecución y autorreflexión.

En el diseño de la presente investigación, estas fases se reflejan explícitamente en las actividades del aula. La previsión se incentiva mediante la observación y el análisis de la obra artística de ejemplo antes de la programación; la ejecución incluye pausas de monitoreo y la autorreflexión se implementa a través de encuestas y bitácoras de aprendizaje que los estudiantes completan al finalizar cada unidad.

3.2.4 Metacognición en el aprendizaje de programación: especificidades del dominio

La programación constituye un entorno propicio para la activación metacognitiva gracias al proceso de depuración de código, o debugging, debido a que, ante un resultado inesperado el programador debe revisar y replantear sus decisiones o el comportamiento de su proyecto.

Este proceso implica acciones como interrumpir o revisar la ejecución, examinar el estado del sistema, identificar diferencias entre el comportamiento del código y el resultado que espera además de formular hipótesis sobre la causa del resultado imprevisto. Estas acciones son comparables en estructura con el modelo de monitoreo y control propuesto por Nelson y Narens (1990). También allí la metodología artística se articula de manera eficiente, pues al trabajar con un referente artístico se identifica un criterio de evaluación específico que facilita la comparación, detección detallada de errores, variaciones o resultados inesperados.

Loksa y Ko (2016) analizaron el proceso de resolución de problemas de programación y observaron que los estudiantes capaces de identificar la etapa del proceso en que se encontraban, obtenían mejores resultados en comparación con aquellos que no lograban una identificación efectiva. Prather et al. (2020) documentan una brecha entre lo que los estudiantes creen que hacen y lo que efectivamente hacen al resolver problemas, y atribuyen la diferencia a un monitoreo poco efectivo antes que a una falta de conocimiento técnico. Los hallazgos de estos estudios son oportunos para interpretar los resultados del Capítulo 6, donde la autorregulación metacognitiva autorreportada disminuye mientras el desempeño técnico permanece en un nivel alto.

El segundo referente relacionado con el área de estudio es el pensamiento computacional. Acciones como descomponer problemas, reconocer patrones, abstraer y diseñar algoritmos son operaciones que obligan a decidir cuándo conviene cambiar de estrategia, en el trasfondo de estas decisiones hay similitudes de naturaleza metacognitiva. Zhang et al. (2025) demostraron que combinar aprendizaje basado en proyectos con programación en parejas mejora a la vez el pensamiento computacional y las habilidades metacognitivas, lo que sugiere que ambos constructos se desarrollan de manera interrelacionada. La revisión de Loksa et al. (2022) sistematiza esta perspectiva y concluye que las intervenciones metacognitivas que demuestran mayor efectividad son aquellas que se integran en el flujo de trabajo del estudiante; en relación con este criterio se diseñó la estructura de las pausas reflexivas descritas en la sección 5.5.

3.2.5 La metacognición como variable mediadora en esta investigación

En el diseño de la presente investigación la metacognición se define como la variable mediadora, es un mecanismo mediante el cual se hipotetiza que las prácticas artísticas generan mejoras en la motivación y el logro de aprendizaje. Estas prácticas crean contextos caracterizados por retroalimentación visual inmediata, una intención estética clara y autonomía creativa, lo que reduce la ansiedad ante el código y fomenta la motivación intrínseca. Simultáneamente, estos contextos estructuran de manera natural los procesos metacognitivos de planificación, monitoreo y evaluación, lo que contribuye a mejorar la calidad del aprendizaje técnico.

3.3 Pilar 3 — Motivación en el aprendizaje de programación

3.3.1 Autoeficacia: la creencia en la capacidad personal como predictor del logro

La autoeficacia, definida por Bandura (1997) como la creencia del individuo en su capacidad para organizar y ejecutar los cursos de acción necesarios para alcanzar determinados resultados, constituye uno de los predictores más sólidos del desempeño académico.

Bandura identificó cuatro fuentes principales para el desarrollo de la autoeficacia: experiencias de maestría, experiencias vicarias, persuasión social y estados fisiológicos y afectivos. La metodología artística propuesta incide en la primera fuente, las experiencias de maestría, mediante el diseño de tareas de creación artística con código que generan resultados visibles y de complejidad creciente.

3.3.2 Teoría de la Autodeterminación: autonomía, competencia y sentido de pertenencia

La Teoría de la Autodeterminación (SDT) de Ryan y Deci (2000) sostiene que la motivación constituye un espectro cualitativo, que va de la amotivación a la motivación intrínseca y comprende diversas formas de motivación extrínseca. Según esta teoría, los entornos educativos que satisfacen las tres necesidades psicológicas básicas, autonomía, competencia y sentido de pertenencia, facilitan la internalización de la motivación. La metodología artística y metacognitiva propuesta aborda esas tres necesidades de forma directa: la elección creativa promueve la autonomía; el logro técnico, observable en el producto artístico, desarrolla la competencia; y la comunidad de práctica que se genera en torno a los proyectos fortalece el sentido de pertenencia.

3.3.3 Teoría de orientación a metas: enfoque en maestría frente a enfoque en rendimiento

La teoría de orientación a metas de Dweck (2000) y Elliot (1999) diferencia entre la orientación hacia la maestría, en la que el estudiante busca aprender y dominar el contenido como objetivo principal, y la orientación hacia el rendimiento, en donde el estudiante aspira a demostrar su competencia ante los demás. En el contexto del aprendizaje introductorio de programación, la orientación motivacional del estudiante influye directamente en su respuesta ante los primeros errores de código. La metodología artística propuesta interviene

en esta dimensión al establecer un entorno donde el error se considera una parte natural y esperada del proceso creativo.

3.4 Pilar 4 — Programación Frontend en primer año universitario

3.4.1 La programación Frontend como dominio específico de aprendizaje

La programación Frontend exige el manejo de lenguajes que permiten construir la capa visible e interactiva de una aplicación web. En el nivel de aprendizaje inicial, ese dominio se estructura en torno a tres tecnologías complementarias: HTML, que define la estructura semántica del contenido; CSS, que determina la presentación visual; y JavaScript, que dota a la página de comportamiento dinámico e interactivo. El aprendizaje del Frontend supone, por ello, una carga cognitiva distribuida: el estudiante debe gestionar al mismo tiempo la sintaxis de cada lenguaje, las relaciones entre ellos y la representación visual que produce el navegador.

Para abordar esta carga cognitiva inicial, el diseño instruccional de la intervención se vale de la librería p5.js como entorno de apoyo en los diferentes módulos del curso. Al eliminar inicialmente las capas de configuración y la sintaxis de HTML y CSS del entorno de codificación, el estudiante puede priorizar sus recursos cognitivos en la comprensión de los temas fundamentales de JavaScript. luego de esto se integran los elementos de HTML y CSS, también se realiza la transición hacia JavaScript puro y las APIs nativas del navegador de manera progresiva.

Park y Wiedenbeck (2011) analizaron el proceso de estudiantes que aprendían desarrollo web y concluyeron que la dificultad principal no reside en la sintaxis de cada lenguaje, sino en el acoplamiento entre ellos y la construcción de un esquema mental del navegador como entorno donde se representa el código. Los participantes resolvían con relativa facilidad tareas aisladas de HTML o CSS, pero presentaban dificultades para anticipar cómo un cambio en una capa podía afectar a las demás. Esta observación coincide con la noción de carga cognitiva propuesta por Sweller (1988) donde la memoria de trabajo del aprendiz es limitada al inicio del curso y el proceso de programación exige mantener simultáneamente tres sistemas de reglas y sus interacciones.

El informe CS2023 (ACM/IEEE-CS/AAAI Joint Task Force, 2024) sitúa los fundamentos del desarrollo de software y los fundamentos algorítmicos en el núcleo obligatorio para todos los graduados, ambos cimientos se abordan en este curso utilizando JavaScript como lenguaje principal (véase sección 5.2). Además de la perspectiva curricular donde los contenidos se relacionan con la formación en computación, la implementación ofrece una posibilidad pedagógica poco explorada: el resultado de cada línea de código es inmediatamente visible, lo que convierte al navegador en un dispositivo de retroalimentación constante.

3.4.2 El problema estructural del abandono en cursos introductorios

Diversos estudios en varios contextos institucionales indican que entre el 30% y el 50% de los estudiantes no logran completar satisfactoriamente su primer curso de programación (Bennedsen & Caspersen, 2007; Robins, Rountree & Rountree, 2003). Entre las posibles causas identificadas se encuentra la elevada carga cognitiva simultánea. También se resaltan algunas problemática asociadas a factores motivacionales como baja autoeficacia inicial, ansiedad frente al código y la utilización de metodologías enfocadas en acciones técnicas que pasan por alto los contextos que son significativos para quienes se encuentra en el proceso de aprendizaje

3.4.3 El contexto latinoamericano

El contexto latinoamericano introduce aspectos particulares al problema donde una de las dificultades identificadas es la distribución desigual de la enseñanza de programación en la educación secundaria (Bocconi et al., 2016; García-Peñalvo, 2016), una proporción significativa de los estudiantes universitarios latinoamericanos accede a los cursos de programación frontend sin experiencia previa en codificación. Paralelamente, el sector del desarrollo web Frontend ha experimentado un crecimiento acelerado en el mercado laboral latinoamericano, lo que otorga a los cursos introductorios una importancia estratégica que va más allá de solamente un cumplimiento curricular (BID, 2020).

3.5 Articulación teórica: el modelo explicativo integrado

Los cuatro pilares teóricos se integran en una cadena de posible causa hipotética entre los 3 eslabones. El primero vincula la metodología artística y metacognitiva, que opera como variable independiente, con la activación metacognitiva que actúa como variable mediadora; las prácticas artísticas se asumen como la estrategia que estructura de manera natural de la implementación, los procesos de planificación, monitoreo y evaluación del aprendizaje. El segundo eslabón relaciona esa activación metacognitiva con los cambios en la motivación, primera variable dependiente: una metacognición efectiva genera experiencias de maestría progresiva, fortalece la autoeficacia (Bandura, 1997) y orienta la motivación hacia la maestría (Dweck, 2000).

En tercer lugar, se conectan los cambios en la motivación con el logro de aprendizaje en Frontend (segunda variable dependiente), en la medida en que una mayor autoeficacia y una orientación a la maestría de los estudiantes sostienen el esfuerzo ante la dificultad y favorecen una comprensión más duradera y transferible en los mismos.

Del modelo explicativo integrado se derivan tres hipótesis de trabajo. En primer lugar se pronostica que la metodología producirá un cambio positivo y estadísticamente significativo en las mediciones de motivación intrínseca y autoeficacia del MSLQ. La segunda anticipa mejoras medibles en el logro de aprendizaje en programación Frontend. La tercera, de carácter mediacional, plantea que los cambios en motivación y logro estarán asociados con un incremento en los indicadores de autorregulación metacognitiva.

Capítulo 4. Metodología

4.1 Tipo y diseño de investigación

Esta investigación adopta un enfoque predominantemente cuantitativo con un componente integrado de orden cualitativo que complementa los resultados numéricos. El diseño es pre-experimental de tipo pre-post con un solo grupo, de modo que las variables dependientes se comparan antes y después de implementar la metodología artística y metacognitiva. Las reflexiones metacognitivas de los estudiantes, el diario docente y las percepciones finales aportan en la profundización y triangulación, a la vez, permiten comprender los mecanismos que subyacen a los cambios cuantitativos observados, estos últimos elementos constituyen el componente cualitativo integrado.

La elección de este diseño responde a las condiciones del contexto de investigación. En primer lugar, no se cuenta con una diferencia entre grupo experimental y grupo de control debido a la cantidad de estudiantes inscritos y matriculados en la asignatura. En segundo lugar, el investigador es al mismo tiempo docente del curso, lo que vuelve inviable conformar un grupo control dentro del mismo sin afectar la equidad y tratamiento pedagógico del grupo. Esta limitación y las implicaciones se reconocen de manera explícita en fase de interpretación de los resultados, las causas y alcances del presente estudio se abordan en la sección de discusión.

4.2 Variables e hipótesis

Tabla 3

Descripción de las variables de estudio

Variable	Tipo	Dimensiones	Instrumento
Metodología artística y metacognitiva	Independiente	Tres módulos artísticos progresivos (pensamiento algorítmico, puntillismo/pixel, arte generativo); secuencia instruccional p5.js → JavaScript puro; estrategias metacognitivas explícitas; alineación con KA CS2023 Cuatro módulos artísticos progresivos (pensamiento algorítmico, puntillismo y píxel digital, abstraccionismo, arte generativo); secuencia instruccional p5.js → JavaScript puro; estrategias metacognitivas explícitas; alineación con las Knowledge Areas del currículo CS2023	Descripción de actividades por módulo
Motivación	Dependiente	Autoeficacia; motivación intrínseca vs. extrínseca; valor de la tarea	MSLQ adaptado al contexto de la presente investigación; aplicación pre y post
Logro de aprendizaje en Frontend	Dependiente	Conocimiento técnico declarativo y procedimental; desempeño aplicado; calidad del producto final	Instrumento integrado de post-test (prueba técnica + then-test retrospectivo + ítems metacognitivos); proyectos entregables
Metacognición	Mediadora	Planificación; monitoreo; evaluación del propio proceso	Subescala de autorregulación metacognitiva del MSLQ

Las hipótesis de trabajo se formulan en correspondencia con el modelo explicativo integrado del Capítulo 3. Cada hipótesis alternativa se acompaña de su hipótesis nula correspondiente, incluida la de carácter mediacional:

- H1: La implementación de la metodología producirá un cambio positivo y estadísticamente significativo en las puntuaciones de motivación intrínseca y autoeficacia del MSLQ (pre-post).

H01: La implementación de la metodología no producirá cambios estadísticamente significativos en las puntuaciones de motivación intrínseca y autoeficacia del MSLQ entre la medición previa y la posterior.

- H2: La metodología producirá mejoras medibles en el logro de aprendizaje en programación Frontend, documentadas a través de la prueba técnica, los proyectos entregables y las rúbricas de desempeño.

H02: La metodología no producirá mejoras medibles en el logro de aprendizaje en programación Frontend, documentables a través de la prueba técnica, los proyectos entregables y las rúbricas de desempeño.

- H3 (mediacional): Los cambios en motivación y logro de aprendizaje estarán asociados con un incremento en los indicadores de autorregulación metacognitiva medidos por el MSLQ y los proyectos de aprendizaje.

H03: Los cambios en motivación y logro de aprendizaje no estarán asociados con variación en los indicadores de autorregulación metacognitiva. Dicho de otro modo, la

autorregulación metacognitiva no operará como variable mediadora entre la intervención y las variables dependientes.

4.3 Población y muestra

La investigación se implementa con estudiantes de primer año universitario inscritos en la asignatura Lenguajes Digitales 2 del programa académico Creación Digital de la Universidad El Bosque, Bogotá, Colombia. Esta asignatura tiene como propósito que los estudiantes aprendan programación Frontend básica a intermedia y opera como el primer curso de profundización técnica en programación para estudiantes que en el semestre anterior cursaron fundamentos de programación, HTML y CSS básico.

El grupo es el grupo natural del aula (muestra por conveniencia), lo que es coherente con el diseño pre-experimental. Se documentarán las características del grupo como variables de contexto para caracterizar la muestra y delimitar la transferibilidad de los resultados.

La muestra efectiva de la investigación varía según el instrumento, en función de la asistencia y la finalización de cada aplicación. El instrumento integrado de post-test fue respondido por $N = 17$ estudiantes. De estos, $N = 13$ estudiantes completaron además ambas aplicaciones del MSLQ (pre y post), constituyendo pares completos válidos para el análisis comparativo intrasujeto de la variable motivación; los 13 estudiantes con MSLQ pre-post completo presentaron también el post-test, por lo que la intersección de estudiantes con datos completos en ambos instrumentos es de $N = 13$, subconjunto usado en los análisis de

correlación entre motivación/metacognición y logro (sección 6.4). Los 4 estudiantes restantes del post-test no cuentan con MSLQ pre-post completo, por decisión voluntaria de no participar en ese instrumento específico; por tanto, sus datos de logro se incluyen sin reservas en los análisis que no requieren cruce con el MSLQ (sección 6.3), y se excluyen únicamente de los análisis que sí lo requieren (sección 6.4).

Para el subconjunto con MSLQ pre-post completo ($N = 13$), las características sociodemográficas fueron las siguientes: la edad promedio fue de 18.69 años ($DE = 0.63$, rango 18–20). En cuanto al género, el 53.8% ($n = 7$) se identificó como masculino, 30.8% ($n = 4$) como femenino, 7.7% ($n = 1$) como no binario y 7.7% ($n = 1$) prefirió no declararlo. En cuanto a la orientación sexual, el 84.6% ($n = 11$) se identificó como heterosexual, 7.7% ($n = 1$) como gay/lesbiana y 7.7% ($n = 1$) prefirió no declararla. Estos datos sociodemográficos no están disponibles para los estudiantes que solo completaron el post-test, dado que se recolectaron únicamente en el instrumento MSLQ.

4.4 Instrumentos de recolección de datos

4.4.1 MSLQ — Instrumento de medición de motivación

El Motivated Strategies for Learning Questionnaire (MSLQ) fue desarrollado por Pintrich, Smith, García y McKeachie (1991) en la Universidad de Michigan, dentro de un programa de investigación sobre el aprendizaje autorregulado. Para esta investigación se trabajó con diez subescalas del instrumento, seleccionadas por su correspondencia directa con las variables dependientes y mediadoras del diseño: seis de la dimensión motivacional (orientación a metas intrínsecas, orientación a metas extrínsecas, valor de la tarea, creencias de control del aprendizaje, autoeficacia para el aprendizaje y el desempeño, y ansiedad ante las pruebas) y cuatro de la dimensión de estrategias de aprendizaje (elaboración, organización, pensamiento crítico y autorregulación metacognitiva). El conjunto suma 55 ítems, se aplicaron en escala Likert de 1 a 10 a través de la plataforma Brightspace y para el análisis, los puntajes se refactorizaron al rango 1–7 del instrumento original (sección 6.2, nota de la Tabla 6).

En cuanto a sus propiedades psicométricas en el contexto colombiano, el instrumento cuenta con una adaptación validada: Ramírez-Echeverry, García Carrillo y Olarte Dussán (2016) tradujeron y adaptaron el cuestionario al español (MSLQ-Colombia) con 1218 estudiantes universitarios colombianos, siguiendo las directrices de la International Test Commission. La validez de constructo se estableció mediante análisis factorial, con índices de adecuación

muestral KMO = .90 para ambas dimensiones del instrumento, y la fiabilidad por consistencia interna arrojó coeficientes α de Cronbach entre .60 y .92 para las subescalas de motivación (por ejemplo, metas intrínsecas = .70, autoeficacia para el desempeño = .81, valor de la tarea = .92) y entre .58 y .84 para las de estrategias de aprendizaje (elaboración = .83, organización = .70, pensamiento crítico = .75). Los autores concluyen que el MSLQ adaptado al contexto Colombiano presenta propiedades psicométricas equivalentes a las del instrumento original en inglés, lo que respalda la pertinencia del MSLQ para población universitaria colombiana como la de este estudio.

4.4.2 Instrumento integrado de post-test

La prueba técnica de logro de aprendizaje combina en un único instrumento tres tipos de datos complementarios: el desempeño técnico objetivo (11 preguntas de selección múltiple y análisis de código), la estimación retrospectiva del conocimiento previo mediante then-test (ítems R y de respuesta retrospectiva por sección) y las percepciones metacognitivas y motivacionales tema por tema (claridad metacognitiva, autoeficacia, aprendizaje percibido, motivación, utilidad de la estrategia y capacidad, en cada sección). En resumen, la estructura del test consta de 99 ítems en total distribuidos en 11 secciones temáticas de 9 ítems cada una. Esta propuesta propia de la intervención constituye una contribución metodológica para explorar con mayor detalle, pues el estudiante responde los ítems de autorreporte inmediatamente después de enfrentar la pregunta técnica correspondiente, lo que permite anclar la reflexión a un referente cognitivo preciso. Conviene precisar que el componente

then-test no estaba previsto en el diseño original del estudio, su implementación se da debido a una contingencia técnica: la evaluación diagnóstica de saberes previos sí se aplicó al inicio del curso a través de la plataforma institucional, pero sus datos se perdieron de forma irrecuperable por un fallo técnico del servidor durante la migración de la plataforma a una nueva versión (sección 5.4). Ante la imposibilidad de repetir una medición inicial, se optó por estimar la línea de base mediante la retrospección estructurada del then-test, asumiendo de manera explícita las limitaciones de interpretación de resultados de esa decisión (secciones 4.6 y 7.4).

4.4.3 Rúbricas de desempeño y portafolio de proyectos

Las rúbricas de desempeño son diseñadas para ser coherentes con los objetivos técnicos del curso y que estén relacionadas con los Knowledge Areas de CS2023 (SDF, AL, GIT).

Adicionalmente, los proyectos entregables por módulo son la evidencia del desempeño del estudiante y de la calidad del producto tanto artístico como técnico que desarrolla.

El diario docente y los registros de sesión documentan la implementación de la metodología con fidelidad al diseño original.

4.5 Procedimiento por fases del diseño

Tabla 4

Fases de implementación

Fase	Momento	Acción	Instrumento
PRE	Inicio del curso	Medición de motivación (autoeficacia e intrínseca/extrínseca) mediante MSLQ y evaluación diagnóstica de saberes previos. Nota: los datos de la evaluación diagnóstica se perdieron por un error técnico del servidor durante la migración de la plataforma (secciones 4.4.2 y 5.4); la estimación de la competencia técnica inicial se realiza retrospectivamente mediante then-test.	MSLQ adaptado (55 ítems, 10 subescalas); evaluación de saberes previos (datos no recuperados)
INTE RVEN CIÓN	Durante el semestre	Implementación de cuatro módulos artísticos con secuencia p5.js → JavaScript puro: (1) Pensamiento algorítmico — Rosas danst Rosas; (2) Puntillismo — del punto al pixel digital; (3) Abstraccionismo — forma y función; (4) Pioneros del arte generativo. Cada módulo sigue el ciclo de cognición encarnada: experiencia corporal → percepción → reflexión → abstracción → codificación → creación. Estrategias metacognitivas: pausas reflexivas, bitácoras de aprendizaje, preguntas de transferencia arte-código.	Diario docente; registros de sesión; entregables por módulo; bitácoras metacognitivas
POST	Cierre del curso	Medición de motivación final; evaluación de logro de aprendizaje técnico; estimación retrospectiva de competencia inicial (then-test); reflexión metacognitiva por tema; percepciones de los estudiantes.	MSLQ (mismas subescalas); Instrumento integrado de post-test (prueba técnica + then-test + ítems metacognitivos);

4.6 Análisis de datos

El análisis de datos se organiza de manera diferenciada según la variable y el instrumento que la mide, dado que la investigación opera con dos situaciones metodológicas distintas en cuanto a la disponibilidad de línea de base.

Para la variable de motivación (MSLQ), el diseño pre-post está completo. El análisis incluye estadísticas descriptivas, con media y desviación estándar por subescala en cada momento, y una prueba de comparación de medias para muestras relacionadas: t de Student pareada o su equivalente no paramétrico, según se verifique o no el supuesto de normalidad mediante la prueba de Shapiro-Wilk, la magnitud del cambio se interpreta con el tamaño del efecto d de Cohen complementa este análisis.

Para la variable de logro de aprendizaje técnico la situación es distinta, puesto que la evaluación diagnóstica inicial se perdió (sección 4.4.2). Este conocimiento técnico previo del estudiante se estima entonces mediante el then-test retrospectivo, el cual está integrado al instrumento de post-test: se calcula la diferencia entre el desempeño objetivo en la pregunta técnica y la competencia previa autopercebida (ítem R), con lo que se construye una variable de ganancia de aprendizaje percibida por tema y por estudiante. Se reconoce de forma

explícita que el then-test produce una estimación frente al estado inicial y no una medida objetiva.

Los ítems de autorreporte metacognitivos y motivacionales por tema (A, M, E, C) se analizan mediante estadísticos descriptivos por sección, lo que permite identificar patrones diferenciales entre los módulos.

4.7 Consideraciones éticas

- La participación en los instrumentos de investigación fue voluntaria e informada, con consentimiento explícito de los estudiantes.
- Los datos fueron anonimizados para el análisis y la publicación de resultados.
- La intervención pedagógica no afectó negativamente a ningún grupo, dado que todos los estudiantes recibieron la metodología y no existió un grupo de control excluido del tratamiento.
- Los resultados académicos de los estudiantes no se vieron afectados por su participación o no participación en los componentes de investigación.

4.8 Uso de inteligencia artificial en el proceso de investigación

Este apartado declara en qué momentos de la investigación se recurrió a asistencia de inteligencia artificial, con qué herramientas y bajo qué controles. La declaración responde a un criterio de transparencia metodológica: el uso de estas herramientas en investigación

educativa exige hoy el mismo nivel de detalle que se le pide a cualquier otro instrumento de trabajo.

4.8.1 Herramientas utilizadas

Se emplearon dos herramientas con asistencia de inteligencia artificial, con funciones distintas y sin solapamiento entre ellas. La primera fue Claude Code, una interfaz de línea de comandos desarrollada por Anthropic, en su versión 2.1.226, que opera sobre modelos de la familia Claude. Se ejecutó localmente en el equipo del investigador, con acceso directo a los archivos del proyecto: las exportaciones de datos de Brightspace, los borradores del documento y las tablas de resultados. No se trató, por tanto, de una conversación abierta con un asistente general, sino de una herramienta que trabajaba sobre los archivos concretos del estudio.

La segunda fue Grammarly, empleada en algunas secciones para la revisión ortográfica y el ajuste de la redacción. Su función se limitó a corregir texto ya escrito: concordancia, puntuación y claridad de la frase. No intervino ni en el contenido ni en la estructura del argumento.

En ningún caso se utilizaron las herramientas como generadoras de contenido. Su papel fue de apoyo a la redacción, con el propósito de dar claridad y precisión a información que ya existía y provenía del propio estudio.

Junto a ellas se emplearon herramientas que no constituyen inteligencia artificial generativa y cuyo papel fue de otra naturaleza: IBM SPSS Statistics 26 y rutinas de Python (pandas, scipy, numpy) para el cálculo estadístico, markdown para la conversión de documentos a texto plano y la interfaz de programación de CrossRef para la verificación de referencias.

4.8.2 Criterio general de uso

El principio que rigió el trabajo fue uno solo: la inteligencia artificial no generó información del estudio, sino que ayudó a expresarla. Toda la información empírica de esta tesis proviene de fuentes que existen con independencia de las herramientas. Los estudiantes respondieron el MSLQ y el post-test integrado en la plataforma institucional, produjeron las 45 entregas que conforman el corpus cualitativo, y las cifras del Capítulo 6 se derivan de esos archivos. El diseño de la intervención, la secuencia de módulos, la elección del puntillismo y el arte generativo como andamiajes y la decisión de aplicar un then-test se tomaron antes de que las herramientas intervinieran en el proyecto.

4.8.3 Uso por momento de la investigación

Momento 1, búsqueda y síntesis bibliográfica: uso parcial. La localización y descarga de las fuentes fue manual y previa (sección 2.1). La asistencia intervino después, en la conversión de los documentos a texto plano, en la organización del corpus por conglomerados y en la redacción de los párrafos de síntesis, siempre sobre los artículos recuperados y no sobre resúmenes de memoria.

Momento 2, tipo de investigación, diseño, variables, hipótesis y objetivos: uso bajo, limitado a la formulación. El diseño pre-experimental respondió a una restricción real del contexto y las variables se escribieron previamente, se sustentaron desde el inicio del proceso de investigación con el marco teórico. La asistencia colaboró en tres tareas puntuales donde su principal función fue sugerir la redacción, estos ajustes de la redacción se realizaron sobre: las hipótesis en su forma convencional, las hipótesis nulas emparejadas y la construcción de objetivos específicos reformulados para que el objeto del verbo fuera la percepción y no la verificación del constructo.

Momento 3, diseño del ambiente de aprendizaje y elaboración de recursos didácticos: sin uso. El curso se diseñó, se dispuso en la plataforma y se implementó antes de que estas herramientas intervinieran en el proyecto. El árbol de módulos, la progresión de p5.js hacia JavaScript sin librerías, las consignas de las actividades artísticas y los materiales del aula son producción del investigador.

Momento 4, recolección de información: sin uso. Los instrumentos se aplicaron en Brightspace y allí respondieron los estudiantes, la asistencia intervino después sobre los archivos ya exportados, su función fue convertir el formato largo en que la plataforma exporta los resultados, en tablas de formato ancho analizables. La clave de respuestas correctas de las once preguntas de conocimiento fue verificada por el investigador.

Momento 5, análisis de los datos: uso medio alto, ligados al procedimiento técnico. Los valores estadísticos no los produjo un modelo de lenguaje sino que fueron aportados por el

software SPSS 26 y se verificaron por medio de scripts de Python. La asistencia escribió los scripts de Python y los procedimientos de SPSS para que el investigador pudiera ejecutarlos de forma independiente. Varios análisis se ejecutaron dos veces (SPSS y por medio de los scripts de Python), luego se compararon los resultados de los procedimientos, de esta manera se permitió detectar y corregir errores en las tablas intermedias, entre ellos un error de ponderación en el puntaje bruto del post-test.

Momento 6, interpretación de los resultados: uso medio, resultados convalidados por el investigador. La asistencia propuso lecturas que fueron aceptadas, corregidas o rechazadas según el caso. La distinción entre medidas de rasgo y medidas de estado que estructura el Capítulo 7 se adoptó por su respaldo teórico y por su ajuste y pertinencia con los datos obtenidos. En sentido contrario, se realizó la exclusión de la pregunta retrospectiva de los análisis comparativos de logro, aunque fueron analizados por la asistencia, se decidió eliminarlos por una observación del investigador sobre el comportamiento del grupo.

Momento 7, sugerencia en la redacción del informe: uso medio-alto. A partir de las tablas verificadas, la asistencia redactó borradores de los capítulos de resultados y discusión, aplicó el formato APA 7 a los reportes estadísticos, construyó la tabla de contenido y las listas de tablas y figuras, además ayudó en la revisión del documento en busca de inconsistencias internas. Sobre algunas secciones cuando ya se contaba con el texto escrito por el investigador y las sugerencias de redacción de la asistencia, se hizo uso de la herramienta

Grammarly con la cual se realizó corrección ortográfica y de estilo, siempre sobre texto ya redactado, nunca se generó datos o texto.

4.8.4 Controles aplicados

En el caso de la verificación de referencias se solicitó apoyo a la asistencia para auditar la bibliografía completa y verificar cada identificador de objeto digital contra la interfaz de programación de CrossRef y comparando el título devuelto con el citado. El procedimiento permitió detectar una referencia inexistente que fue eliminada y sustituida por fuentes verificadas; en los casos en que un identificador no se resolvió correctamente, la referencia se dejó sin enlace en lugar de asignarle uno.

Trazabilidad de decisiones. Cada decisión significativa del estudio quedó registrada con fecha y justificación en un archivo de seguimiento del proyecto, el cual permite reconstruir y hacer seguimiento a las elecciones y evita contradicciones entre sesiones de trabajo.

Verificación de cifras contra la fuente. Los valores de las tablas del Capítulo 6 se contrastaron uno por uno con los archivos de datos originales, los cuales se guardaron en formato csv, de esta manera, la comprobación se realizó de forma independiente sobre los archivos csv y no sobre lo registrado en la conversación de trabajo.

Anonimización. Los estudiantes citados en el análisis cualitativo aparecen identificados mediante códigos. La correspondencia entre nombre y código se conserva en un archivo interno que no forma parte de este documento.

4.8.5 Alcance y responsabilidad

El investigador asume la responsabilidad completa sobre el contenido de este documento.

Los datos son reales, las cifras son verificables contra los archivos fuente del proyecto y las referencias fueron comprobadas contra bases externas. Las decisiones metodológicas, las interpretaciones y los errores que puedan subsistir corresponden al investigador.

En el Anexo H se relacionan ejemplos de las instrucciones dadas a la herramienta organizados según los siete momentos descritos en este apartado.

En línea con las directrices de la American Psychological Association sobre el uso de sistemas generativos en producción académica se argumenta que ninguna herramienta figura como autora ni coautora de la presente investigación.

Capítulo 5. Propuesta Pedagógica: el Curso como Intervención

5.1 Diseño instruccional: filosofía y principios

La propuesta pedagógica que constituye la variable independiente de esta investigación, se implementó en el curso Lenguajes Digitales 2, del programa de Creación Digital de la Universidad El Bosque, el cual se ofrece en el segundo semestre del programa académico. En este nivel se introduce a los estudiantes en la programación web interactiva a través de la exploración de corrientes artísticas que fueron pioneras en el desarrollo de herramientas digitales. Se fundamenta, además, en el principio de cognición encarnada, según el cual el conocimiento emerge de la interacción dinámica entre el sujeto, su cuerpo y su entorno.

El ciclo epistemológico que hace parte de la implementación se estructura de manera secuencial. Inicialmente, la experiencia corporal da paso a la percepción, luego a la reflexión, seguida de la abstracción, continuando por la implementación y finalmente la expresión por medio del código.

De esta manera los estudiantes perciben, tocan, observan y discuten una situación, una obra o un fenómeno artístico antes de realizar la implementación por medio del código. El error, por su parte, se trata explícitamente como recurso creativo y activador de la metacognición, contrario a una señal de fracaso.

5.2 Alineación curricular con CS2023

El diseño instruccional de la implementación se elaboró en coherencia con el modelo de conocimiento del currículo CS2023 (ACM/IEEE-CS/AAAI Joint Task Force, 2024). En conjunto, los cuatro módulos temáticos activan 9 de las 17 Knowledge Areas del CS2023, cuatro de ellas pertenecientes al CS Core obligatorio para todo graduado en computación.

Tabla 5
Alineación curricular

Módulo	Referente artístico	Contenidos técnicos	KA CS2023 activadas
1 — Pensamiento algorítmico	Rosas danst Rosas (De Keersmaeker, 1983)	Pseudocódigo, ciclos (for, while, do-while), variables, condicionales	AL (CS Core), SDF (CS Core), MSF (CS Core)
2 — Puntillismo / Pixel digital	Seurat, Signac	Tipos de dato, modelo RGB, sistema de coordenadas, loops anidados, aleatoriedad controlada con p5.js	SDF (CS Core), GIT (KA Core), MSF (CS Core)
3 — Puntillismo / Abstraccionismo	Kandinsky, Mondrian	Funciones con parámetros, arrays, objetos, métodos, DOM, CSS	SDF (CS Core), FPL (KA Core), HCI (KA Core), SPD (KA Core)
4 — Arte generativo	Molnár, Nees, Mohr, Reas	Algoritmos generativos, Canvas API, event listeners.	AL (CS Core), GIT (KA Core), AI (CS Core), SPD (KA Core), SEP (transversal)

5.3 Progresión de herramientas: de p5.js a JavaScript

La implementación es elaborada con base a la experiencia como artista, docente y programador por parte del investigador. En los módulos 1 y 2, se utiliza p5.js como entorno de andamiaje gracias a su sintaxis simplificada basada en Javascript y la posibilidad de

retroalimentación visual inmediata.

Esta característica de implementación es consistente con la reducción de la carga cognitiva extrínseca (Sweller, 1988) pues el estudiante construye un modelo mental funcional de JavaScript antes de enfrentarse a la complejidad del sistema integrado del navegador.

Desde el módulo 3 y hasta el módulo 4, la instrucción migra de forma progresiva hacia JavaScript puro, el DOM nativo, los event listeners y la Canvas API. El referente artístico aparece en todos los módulos al inicio como elemento de referencia y contexto: el tránsito tecnológico está siempre mediado por una intención creativa que da sentido al aprendizaje técnico.

5.4 El ambiente de aprendizaje en Brightspace (AVA)

La propuesta pedagógica se materializó en un curso alojado en Brightspace, el sistema de gestión del aprendizaje institucional del que dispone la Universidad El Bosque. El ambiente virtual cumplió una doble función en esta investigación: por un lado fue el entorno de implementación de la intervención, donde los estudiantes accedieron a los contenidos, realizaron las actividades de creación y entregaron sus productos, simultáneamente, fue el vehículo de recolección de todos los instrumentos del estudio. La descripción que sigue se basa en el paquete de exportación oficial del curso en formato IMS Content Package, que registra la estructura completa del ambiente: 18 módulos y submódulos de contenido, 31 páginas de recursos, 12 encuestas, 8 carpetas de entrega, un foro de discusión, una rúbrica de

evaluación y un objeto de contenido interactivo H5P. La Figura 1 presenta la estructura general del ambiente tal como la veía el estudiante.

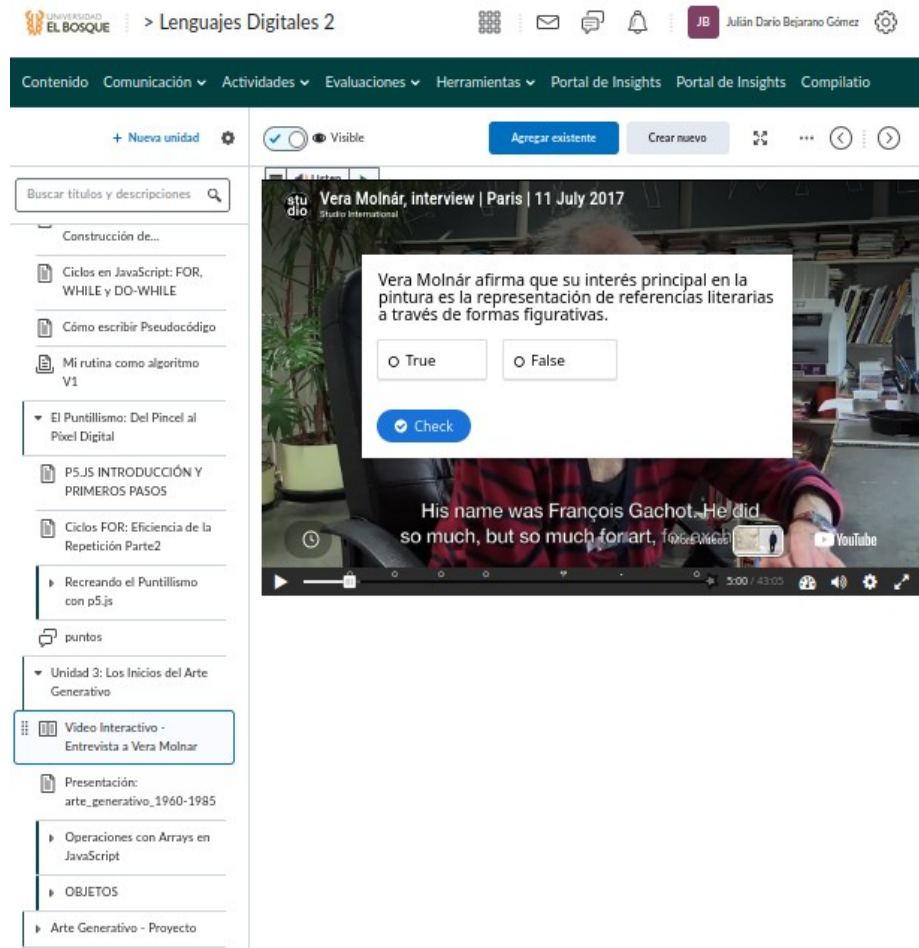
La estructura de módulos del curso replica la secuencia instruccional descrita en las secciones 5.2 y 5.3. Tras un módulo de bienvenida con el syllabus y la presentación del docente, el módulo *Rosas danst Rosas — Pensamiento Algorítmico* despliega los contenidos de pseudocódigo y ciclos en JavaScript. El módulo *El Puntillismo: Del Pincel al Píxel Digital* introduce p5.js y culmina en la actividad de creación de la obra puntillista. La unidad *Los Inicios del Arte Generativo* articula el referente histórico (una entrevista interactiva a Vera Molnár construida en H5P, Figura 2, y una presentación sobre el arte generativo de 1960–1985) con los bloques técnicos de arrays y objetos en JavaScript. Por último, el módulo *Arte Generativo — Proyecto* concentra las guías pedagógicas, la plantilla del proyecto final integrador y ejemplos de código en p5.js que recrean obras de los cuatro pioneros trabajados en el curso (Molnár, Nees, Nake y Mohr). La organización mantiene el principio fundamental de la propuesta en donde cada módulo comienza con un referente artístico que sirve de contexto y guía para el contenido técnico.

Estructura general del Ambiente Virtual de Aprendizaje



Figura 2

Ambiente Virtual de Aprendizaje (sección inicios del arte generativo, video interactivo Vera Molnar)

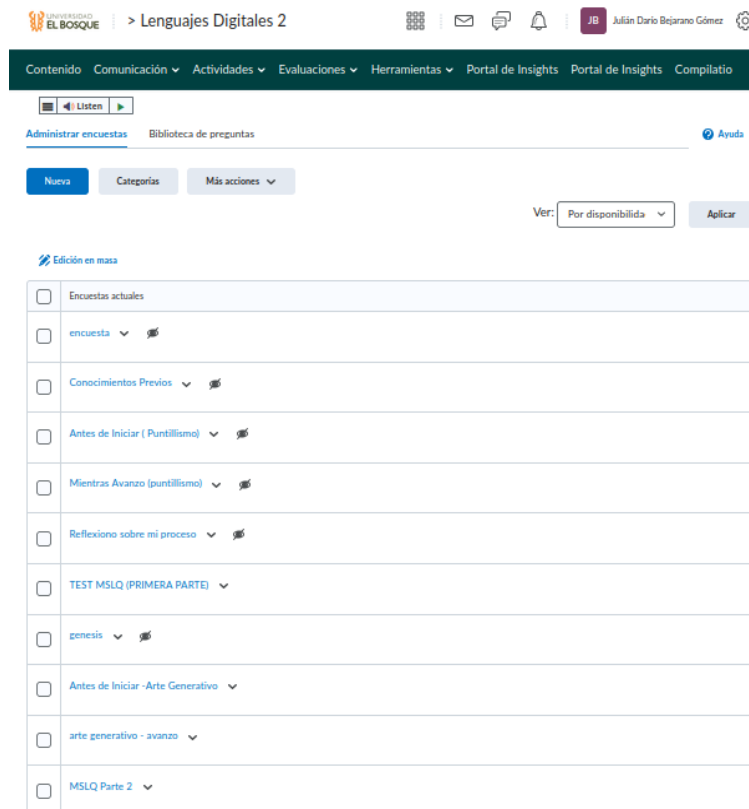


Un rasgo de diseño distintivo del ambiente es la integración de las encuestas metacognitivas dentro de la secuencia de contenidos. En las secciones de puntillismo y arte generativo se disponen encuestas para tres momentos: una encuesta previa a la actividad ("Antes de Iniciar"), en la que el estudiante anticipa la dificultad y anticipa su plan; luego, una encuesta

durante la ejecución ("Mientras Avanzo"), que interrumpe a proposito el trabajo para activar la fase de monitoreo; y finalmente una encuesta posterior ("Reflexiono sobre mi proceso" / "Evalúo mi proceso"), de autoevaluación retrospectiva. Estas encuestas las encuentra el estudiante en el flujo natural de navegación de los módulos. La Figura 3 muestra la ubicación de estas encuestas dentro de la secuencia del módulo.

Figura 3

Sección de encuestas del Ambiente Virtual de Aprendizaje



Este diseño aplica en la plataforma el ciclo de autorregulación propuesto por Zimmerman (2000), descrito en la sección 3.2.3, adicionalmente, constituye la fuente directa de los datos analizados en las secciones 6.6 a 6.8.

El componente de evaluación dentro del ambiente se organizó en torno a entregas iterativas de algunas actividades, es decir, entregas en versiones sucesivas (*Mi rutina como algoritmo* V1 y V2; *Implementación de Objetos* en versión inicial e intermedia), de modo que el estudiante recibe retroalimentación sobre una primera versión y luego hace el envío de una versión mejorada. Esa mecánica resulta coherente con el tratamiento del error como recurso creativo (sección 5.1).

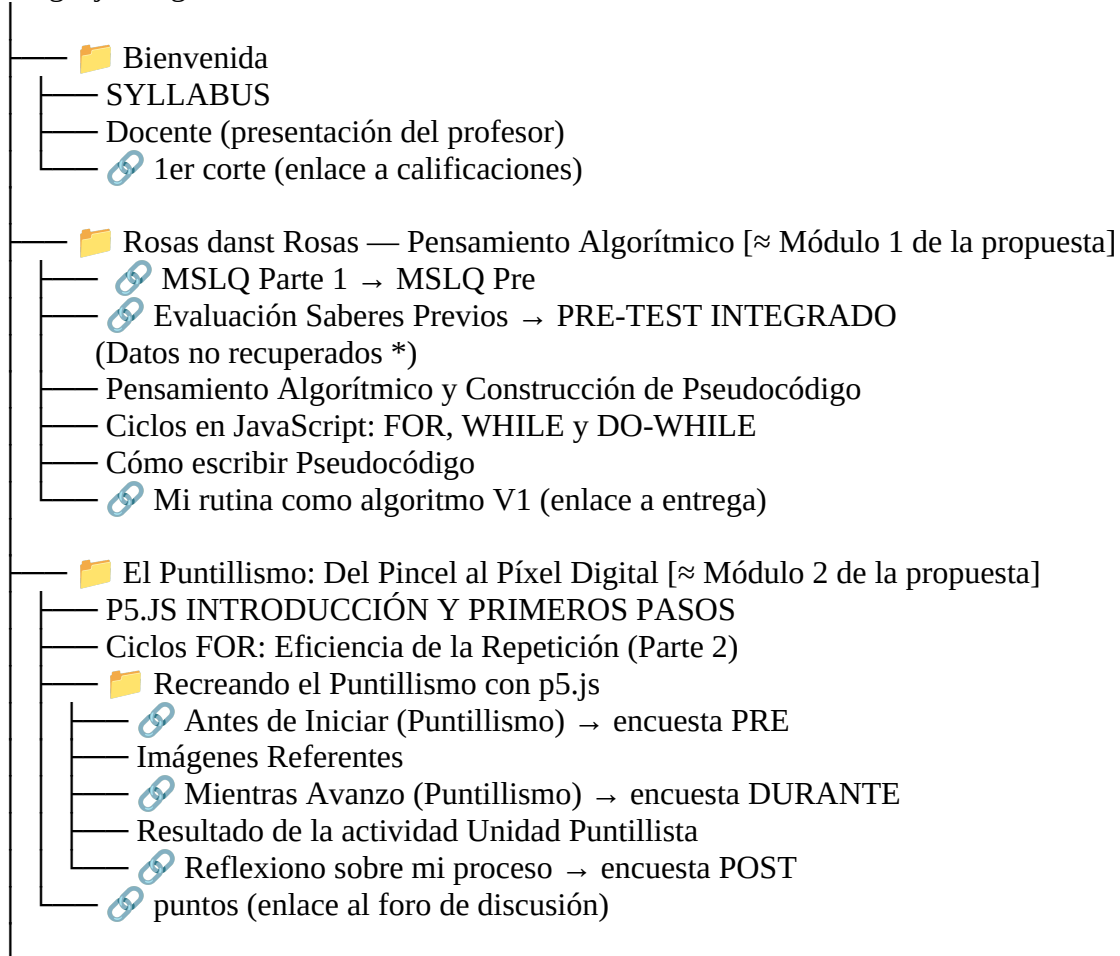
El proyecto final integrador (*Tu Primer Sistema Generativo*, Front-end + p5.js) cierra la secuencia de entregas. A través de la misma plataforma se aplicaron los instrumentos de investigación: la encuesta de conocimientos previos al inicio del curso, el MSLQ en sus dos momentos (pre y post), las encuestas pre, durante y post de las prácticas de Puntillismo y Arte Generativo, más el post-test integrado de 99 ítems (11 secciones temáticas con 9 ítems cada una), aplicado al cierre del semestre. Las calificaciones se organizaron en los tres cortes académicos institucionales.

Por último, y en coherencia con la Dimensión 1 del seguimiento docente (fidelidad de implementación, sección 5.6), cabe mencionar una variación entre el diseño original y la implementación. El módulo de Abstraccionismo previsto en el primer diseño (sección 5.2) no se desplegó como módulo independiente en la plataforma, sin embargo, sus contenidos

artísticos y técnicos (funciones, arrays y objetos) se integraron dentro de la unidad de los inicios del arte generativo. La variación no altera la secuencia p5.js → JavaScript puro ni el alcance propuesto de contenidos, este cambio de diseño implicó que el abstraccionismo tuvo menor presencia estructural en el ambiente en comparación con los otros tres referentes y fue elegido para lograr una coherencia con el número de cortes académicos (3 en vez de 4).

Árbol de módulos y contenidos

Lenguajes Digitales 2



—	📁	Unidad 3: Los Inicios del Arte Generativo [≈ Módulos 3-4 de la propuesta]
—	🎬	Video Interactivo — Entrevista a Vera Molnár (H5P)
—		Presentación: arte_generativo_1960-1985
—	📁	Operaciones con Arrays en JavaScript
—		Fundamentos de los Arrays
—		Buscar Elementos en un Array
—		Eliminar e Insertar Elementos
—		Ejercicios de tarea
—		Código: Clase Gestor de Listas
—	📁	OBJETOS
—	📦	BLOQUE 1 — Objetos en JavaScript
—	📁	BLOQUE 2 — Arrays de Objetos
—		BLOQUE 3 — for & forEach con objetos
—		BLOQUE 4 — Template Literals
—		Implementación de Objetos — Versión Inicial (entrega)
—	🔗	Implementación de Objetos — Versión Intermedia (entrega)
—	📁	Arte Generativo — Proyecto [≈ Módulo 4 de la propuesta]
—		Links de referencia a ejercicios en CodePen
—		Guía pedagógica: Arte Generativo (completa)
—		Plantilla: Proyecto Final Integrador — Tu Primer Sistema Generativo
—		storytelling-ld2
—		Guía Pedagógica: Cuadrados Concéntricos con Perturbación
—	📁	Ejemplos de código: Arte Generativo en p5.js
—		mohr_art (Manfred Mohr)
—		nees_art (Georg Nees)
—		nake1 (Frieder Nake)
—		molnar_art (Vera Molnár)
—	📁	Encuestas y evaluaciones
—	🔗	Antes de Iniciar — Arte Generativo → encuesta PRE
—	🔗	arte generativo — avance → encuesta DURANTE
—	🔗	Arte Generativo — Evalúo mi proceso → encuesta POST
—	🔗	MSLQ Parte 2 → MSLQ POST
—	🔗	Evaluación 28 de Mayo → POST-TEST INTEGRADO (99 ítems)
—	📁	STORYTELLING DE ESTE CURSO (sin sub-contenidos en el export)
—	📁	30 Ejercicios JS (sin sub-contenidos en el export)
—	📁	Repaso Semana de Receso (sin sub-contenidos en el export)
—	📁	POESIA ALGORITMICA (sin sub-contenidos en el export)
—	📁	PruebasRecursos (módulo de pruebas del docente, no pedagógico)

* Estos datos se perdieron por inconvenientes en el servidor durante la migración a una nueva versión de la plataforma.

5.5 Estrategias metacognitivas integradas

Tres dispositivos pedagógicos explícitos integran la metacognición al flujo de trabajo de cada módulo. Las pausas reflexivas estructuradas detienen el trabajo de codificación para que el estudiante responda preguntas orientadas al monitoreo: ¿qué estoy intentando lograr?, ¿mi código está haciendo lo que creo que hace?, ¿en qué punto exacto estoy atascado? Las bitácoras de aprendizaje, por su parte, son registros escritos al cierre de cada sesión en los que el estudiante documenta qué aprendió, qué le resultó difícil, qué estrategia usó para superar la dificultad y qué haría distinto la próxima vez. Las preguntas de transferencia arte-código, finalmente, hacen consciente el puente entre la intención estética, como elegir un tipo de curva porque produce una sensación de calma, y la decisión técnica que la implementa.

5.6 Dimensiones de observación y seguimiento docente

Se registran cuatro dimensiones de seguimiento a lo largo del semestre orientadas a documentar la fidelidad de implementación y ajustar el plan de trabajo de manera progresiva:

- Dimensión 1. Implementación de la metodología: qué prácticas artísticas y metacognitivas se implementaron, con qué nivel de fidelidad al diseño original y qué variaciones se introdujeron.
- Dimensión 2. Señales de motivación: cómo reaccionaron los estudiantes ante las dificultades técnicas, si hubo iniciativa propia o personalización creativa, si la energía del grupo varió con el componente artístico.
- Dimensión 3. Señales de logro de aprendizaje: qué habilidades técnicas emergen, qué errores son recurrentes y qué revelan sobre la comprensión.
- Dimensión 4. Procesos metacognitivos: si los estudiantes identifican qué entienden y qué no, si ajustan su estrategia cuando algo no funciona, si las reflexiones escritas muestran profundidad.

Capítulo 6. Resultados y Análisis

Este capítulo presenta los resultados cuantitativos y cualitativos de las tres variables centrales del estudio (motivación, logro de aprendizaje y metacognición), de acuerdo con el análisis de datos descrito en el Capítulo 4. La sección 6.5 presenta el análisis cualitativo de las bitácoras de aprendizaje de tres actividades del curso; los campos de reflexión abierta del instrumento integrado de post-test no se incluyen en este corte del análisis (sección 7.4).

6.1 Verificación de supuestos estadísticos

Los análisis estadísticos reportados en este capítulo se ejecutaron con el software IBM SPSS Statistics, versión 26 (IBM Corp., 2019).

Antes de comparar las medias de la variable motivación se verificó el supuesto de normalidad con la prueba de Shapiro-Wilk, aplicada a las 10 subescalas del MSLQ en ambos momentos de medición (escala refactorizada 1–7), sobre los $N = 13$ estudiantes con pares completos pre-post. De las 20 pruebas realizadas ($10 \text{ subescalas} \times 2 \text{ momentos}$), 18 no mostraron evidencia de desviación de la normalidad ($p > .05$). Solo dos distribuciones incumplieron el supuesto: Valor de la Tarea en el momento pre ($W = .864, p = .044$) y Creencias de Control del Aprendizaje en el momento pre ($W = .841, p = .022$).

Dado este cumplimiento parcial, y puesto que un tamaño muestral reducido ($N = 13$) limita la potencia de Shapiro-Wilk para detectar desviaciones de la normalidad de forma confiable, se optó por reportar para las 10 subescalas tanto la prueba t de Student para muestras

relacionadas como su equivalente no paramétrico, la prueba de rangos con signo de Wilcoxon, en lugar de seleccionar un único estadístico por subescala. Ese procedimiento de doble verificación permite evaluar la consistencia de los hallazgos y tratamiento de datos frente a la violación parcial del supuesto.

6.2 Variable de motivación: resultados MSLQ pre-post

Tabla 6
Comparación pre-post de las subescalas MSLQ (escala 1–7), N = 13

Subescala	M Pre (DE)	M Post (DE)	t(12)	p (t)	dz	Z (Wilcoxon)	p (W)	r
IM — Metas intrínsecas	4.98 (1.78)	5.08 (1.36)	-0.55	.596	-0.15	-0.93	.350	.26
EM — Metas extrínsecas	4.81 (1.24)	4.51 (1.14)	1.15	.272	0.32	-0.79	.431	.22
VT — Valor de la tarea	5.49 (1.57)	5.50 (1.12)	-0.02	.987	-0.01	-0.36	.722	.10
CL — Creencias de control	5.35 (1.04)	5.13 (0.80)	0.76	.464	0.21	-1.14	.255	.32
AE — Autoeficacia	5.19 (1.45)	5.06 (1.08)	0.49	.630	0.14	-0.52	.600	.14

AN — Ansiedad	4.15 (0.86)	4.26 (1.04)	-0.35	.732	-0.10	-0.12	.906	.03
EL — Elaboración	4.57 (1.25)	4.51 (0.98)	0.20	.841	0.06	-0.31	.753	.09
OR — Organización	3.80 (1.47)	4.05 (1.23)	-0.99	.342	-0.27	-0.86	.387	.24
PC — Pensamiento crítico	4.57 (1.82)	4.85 (1.43)	-0.63	.539	-0.17	-0.45	.657	.12
MR — Autorregulación metacognitiva	4.65 (0.91)	4.19 (0.99)	2.22	.046	0.62	-1.92	.055	.53

Nota. $dz = d$ de Cohen para muestras relacionadas (signo positivo = descenso post-intervención; signo negativo = incremento post-intervención); r = tamaño del efecto de Wilcoxon ($Z/\sqrt{N}$). Escala refactorizada a rango 1–7 mediante la transformación $0.667 \times (M - 1) + 1$ a partir del puntaje original en escala 1–10.

De las diez subescalas evaluadas, únicamente la Regulación Metacognitiva (MR) mostró un cambio relevante entre los momentos pre y post: un descenso estadísticamente significativo en la prueba t pareada, $t(12) = 2.22$, $p = .046$, $dz = 0.62$ (efecto mediano-alto), y un resultado marginal mediante Wilcoxon, $Z = -1.92$, $p = .055$, $r = .53$ (efecto grande). Ambas pruebas coinciden en dirección y en magnitud, incluso con la diferencia en la significancia

estadística se respalda la interpretación de un cambio real y no de un artefacto del método estadístico empleado.

Ninguna otra subescala alcanzó significación estadística en las dos pruebas, con tamaños de efecto pequeños a moderados (r entre .03 y .32). En particular, las dos subescalas centrales de la Hipótesis 1, Orientación a Metas Intrínsecas (IM) y Autoeficacia (AE), no mostraron cambios significativos: IM presentó un leve incremento no significativo ($t(12) = -0.55$, $p = .596$) y AE un leve descenso, tampoco significativo ($t(12) = 0.49$, $p = .630$). Estos resultados no ofrecen soporte estadístico para H1 teniendo en cuenta el tamaño de la muestra ($N = 13$ pares completos).

El hallazgo más relevante del bloque es el descenso en regulación metacognitiva. Ese resultado contradice la lógica del modelo explicativo integrado, que anticipa un fortalecimiento de los procesos metacognitivos por efecto de las prácticas artísticas. Una interpretación viable, que se desarrolla en la discusión (Capítulo 7), es que el descenso no refleje una pérdida real de autorregulación, sino un ajuste del punto de referencia del estudiante: tras experimentar de forma directa el proceso de programar con retroalimentación visual inmediata y responder los test distribuidos a lo largo del semestre, es posible que los estudiantes hayan desarrollado un criterio más exigente o más realista, sobre lo que significa monitorear y regular el propio aprendizaje, y que en consecuencia se autoevalúen con mayor rigurosidad en el post-test que en el pre-test.

6.3 Variable de logro de aprendizaje: resultados del instrumento integrado

El instrumento integrado de post-test fue respondido por $N = 17$ estudiantes en 11 secciones temáticas activas. El puntaje total ponderado, cuyo máximo es 13 puntos (S12 y S15 valen 2 puntos cada una y las 9 secciones restantes, 1 punto), obtuvo una media de $M = 10.59$ ($Mdn = 11.00$, $DE = 2.00$, rango 7–13), equivalente a una calificación en escala 0–5 de $M = 4.07$ ($DE = 0.77$).

Tabla 7

Porcentaje de respuestas correctas por sección temática ($N = 17$)

Sección	Tema	% correctas
S1	Tipos de dato en JavaScript	100.0
S6	Arrays — acceso por índice	94.1
S16	DOM — getElementById / addEventListener	94.1
S15	Análisis de código — Schotter (desorden progresivo)	88.2
S2	Condicionales (if / else if / else)	82.4
S7	Métodos de array (filter())	82.4
S10	Arte generativo y pioneros	82.4
S4	Funciones y valor de retorno	76.5
S12	Análisis de código — map()	76.5
S17	DOM — value e innerHTML	70.6
S3	Loop for (p5.js)	47.1

El desempeño más bajo se presentó en la Sección 3, correspondiente al ciclo for en contexto p5.js, con el 47.1% de aciertos. Contrario a este resultado, los tipos de dato (S1) y el acceso a arrays por índice y al DOM (S6, S16) mostraron un dominio prácticamente completo: 100% y 94.1%, respectivamente.

Con respecto a los ítems de autorreporte por tema (metacognición, autoeficacia, conocimiento retrospectivo, aprendizaje percibido, motivación, utilidad de la estrategia y capacidad), se calculó la media de cada categoría a través de las 11 secciones:

Tabla 8

Descriptivos de las medias por categoría de autorreporte del post-test (escala 1–5, N = 17)

Variable	M	Mdn	DE	Min	Max
Metacognición — claridad del tema (MC)	3.10	3.00	0.70	2.00	5.00
Autoeficacia (AE)	3.20	3.09	0.72	2.00	5.00
Conocimiento previo retrospectivo (R)	1.96	2.00	0.62	1.09	3.09
Aprendizaje percibido (A)	3.12	3.09	0.75	1.91	5.00
Motivación (M)	3.18	3.18	0.85	2.00	5.00
Utilidad de la estrategia (E)	3.18	3.09	0.80	2.09	5.00
Capacidad (C)	3.05	3.00	0.76	2.00	5.00

Nota: el ítem P9 (then-test retrospectivo) fue excluido del análisis comparativo de logro. Se observó que varios estudiantes que no dominaban el tema seleccionaron la opción correcta en el then-test en lugar de reconocer "No sé / No recuerdo", esto sugiere contaminación por adivinanza retrospectiva o por deseabilidad social antes que una medida fiable del conocimiento previo real. La ganancia de aprendizaje percibida se tomó como la diferencia

entre la autoeficacia declarada (P3-AE) y el conocimiento previo autorreportado (P4-R), ambos hacen parte del mismo instrumento integrado en el post-test.

Se calcularon correlaciones de Spearman entre el puntaje total del post-test y tres indicadores autorreportados: autoeficacia percibida, claridad metacognitiva y ganancia de aprendizaje percibida.

Tabla 9

Correlaciones de Spearman entre indicadores percibidos y logro real (N = 17)

Variable	ρ	p
Autoeficacia percibida (AE)	.35	.165
Claridad metacognitiva (MC)	.29	.252
Ganancia de aprendizaje percibida	.58	.014
(AE – Retrospectivo)		

La ganancia de aprendizaje percibida mostró una correlación positiva, moderada-fuerte y estadísticamente significativa con el logro real, $\rho(15) = .58$, $p = .014$. Quienes reportaron una mayor diferencia entre su autoeficacia actual y su conocimiento previo mostraron tendencia a obtener puntajes más altos en el post-test. Ni la autoeficacia percibida por sí sola, $\rho(15) = .35$, $p = .165$, ni la claridad metacognitiva percibida, $\rho(15) = .29$, $p = .252$, alcanzaron significación estadística respecto del logro medido.

La percepción de avance (más que la autoeficacia o la claridad conceptual vistas de manera aislada), parece asociarse más consistentemente con el desempeño técnico observado, aunque el tamaño muestral reducido exige cautela interpretativa. Al igual que en la Tabla 10, estas tres comparaciones no incluyen corrección por comparaciones múltiples: al tratarse de solo tres pruebas planificadas y no de diez, el riesgo de falsos positivos es menor. Se menciona por consistencia metodológica con el resto del capítulo.

6.4 Variable mediadora: metacognición

Como se reportó en la sección 6.2, la subescala de Autorregulación Metacognitiva (MR) del MSLQ fue la única que mostró un cambio estadísticamente relevante entre el pre y el post, con un descenso de $M = 4.65$ a $M = 4.19$ (escala 1–7), $t(12) = 2.22$, $p = .046$, $dz = 0.62$; Wilcoxon $Z = -1.92$, $p = .055$, $r = .53$.

Para examinar el papel de MR como posible variable mediadora entre la intervención y el logro (H3), se calculó la correlación de Spearman entre MR en el momento post y el puntaje total del post-test integrado, sobre el subconjunto de estudiantes con datos completos en ambos instrumentos ($N = 13$, la totalidad de quienes cuentan con MSLQ pre-post completo). El mismo análisis se extendió a las nueve subescalas restantes con el fin de contextualizar el hallazgo.

Tabla 10

Correlación de Spearman entre subescalas MSLQ (post) y puntaje total del post-test

<i>integrado (N = 13)</i>		
Subescala	ρ	p
EL — Elaboración	.70	.008
OR — Organización	.63	.021
CL — Creencias de control	.61	.027
IM — Metas intrínsecas	.54	.059
AE — Autoeficacia	.53	.062
PC — Pensamiento crítico	.49	.086
MR — Autorregulación metacognitiva	.47	.104
VT — Valor de la tarea	.47	.106
EM — Metas extrínsecas	.38	.195
AN — Ansiedad	-.09	.778

Contra lo planteado en la hipótesis, MR no resultó ser la subescala más asociada con el logro de aprendizaje: su correlación con el puntaje total, $\rho(11) = .47$, $p = .104$, no alcanzó significación estadística en esta submuestra ($N = 13$), aunque la magnitud del efecto es moderada. En cambio, tres subescalas mostraron correlaciones significativas con el logro real: Elaboración (EL, $\rho = .70$, $p = .008$), Organización (OR, $\rho = .63$, $p = .021$) y Creencias de Control del Aprendizaje (CL, $\rho = .61$, $p = .027$). Orientación a Metas Intrínsecas (IM) y Autoeficacia (AE) se ubicaron en un nivel marginal, sin alcanzar el umbral de significancia de .05 ($p = .059$ y $p = .062$, respectivamente). Estos resultados exigen prudencia en su interpretación debido a que se realizaron diez comparaciones sin corrección por

comparaciones múltiples sobre una muestra de 13 estudiantes, de modo que el riesgo de falsos positivos es considerable y la potencia estadística baja.

En lugar de confirmar la hipótesis de mediación inicial (H3) de forma definitiva, estos datos sugieren que la relación es más detallada y llena de matices: la autorregulación metacognitiva medida por el MSLQ (MR) disminuyó de forma significativa tras la intervención, pero no es la subescala que mejor predice el logro de aprendizaje. En cambio, son las estrategias de aprendizaje: elaboración y organización, junto con las creencias de control del aprendizaje, las que muestran una asociación más consistente con el desempeño técnico observado. Ello no descarta un papel mediador de la metacognición en sentido amplio, si bien indica que ese papel podría operar a través de procesos de elaboración y organización de la información antes que a través de la autorregulación metacognitiva tal como la mide la subescala MR. El análisis de contenido de las bitácoras de aprendizaje (sección 6.5) ofrece una primera lectura cualitativa de este hallazgo; la triangulación con los campos de reflexión abierta del instrumento integrado de post-test, no incluidos en este corte, queda como línea de trabajo futuro (sección 7.7).

6.5 Resultados cualitativos

Se realizó un análisis de contenido temático de las bitácoras de aprendizaje producidas en tres actividades del curso: Métodos en Arreglos (N = 17), Mi Primera Obra Puntillista Digital (N = 13) y Mi Primer Sistema Generativo (N = 14). Este análisis cubre las bitácoras de aprendizaje descritas en la sección 5.5.

La actividad Métodos en Arreglos incluye tres preguntas fijas de reflexión que corresponden de forma directa a las tres fases del ciclo de autorregulación de *Zimmerman (2000, sección 3.2.3)*: previsión (¿qué estrategia usarás y por qué?), ejecución (¿por dónde empezaste, qué decisiones tomaste?) y autorreflexión (¿el resultado fue el esperado?, ¿qué cambiarías?). Las otras dos actividades no incluyen preguntas de reflexión estructuradas de la misma forma, si bien sus entregas contienen justificaciones de diseño, en particular la sección “Control vs. Aleatoriedad”, que constituyen actos de regulación metacognitiva aplicados al dominio artístico-técnico. La codificación combinó un marco deductivo (las tres fases de *Zimmerman*) con categorías inductivas emergentes de la lectura completa del corpus (Tabla 11). La realizó un solo analista, el propio investigador, sin verificación de fiabilidad entre codificadores, por lo que las frecuencias deben leerse como una aproximación cualitativa, no como un resultado estadístico. Los nombres de estudiantes se reemplazaron por códigos (E01-E19), siguiendo el compromiso de anonimización de este estudio (sección 4.7).

Tabla 11

Categorías emergentes en las reflexiones y entregas de estudiantes, por fase del

<i>ciclo de autorregulación (Zimmerman, 2000)</i>				
Fase	Código	Categoría	n	Cita ilustrativa
Previsión	P1	Estrategia por reconocimiento funcional	15/17	“Elegí usarlos porque sirven para agregar al principio (unshift), agregar al final (push) y quitar elementos (pop)” (E04)
Previsión	P2	Recurso a andamiaje externo como estrategia declarada	5/17	“Tuve que recurrir al uso de inteligencia artificial... no tenía idea de dónde iniciar” (E01)
Previsión	P3	Elección sin criterio técnico explícito	3/17	“Lo elegí al azar entre los que faltaban” (E02)
Previsión	P4	Transferencia estética-técnica	10/14	“El desorden tiene que venir del tamaño y la posición, no de la rotación” (E16)
Previsión	P5	Distinción explícita control/azar como decisión de diseño	4/14	“Si todo es azar, no hay sistema. Hay que decidir qué controlas y qué dejas al azar” (E16)
Ejecución	E1	Verificación incremental (monitoreo activo)	10/17	“Fui mostrando todo con console.log() para ver que sí estuviera funcionando” (E06)
Ejecución	E2	Detección de error y autocorrección en curso	3/17	“Intenté usar un for para sumar los elementos, pero me di cuenta de que no funcionaba porque eran palabras y no números. Entonces cambié la forma” (E04)

Ejecución	E4	Apoyo externo ante bloqueo	3/17	“El código estaba teniendo muchos errores... lo cambié por indexOf con ayuda de una IA” (E02)
Autorreflexión	R1	Autoevaluación con transferencia a la práctica futura	7/17	“Primero pensaría mejor qué tipo de datos estoy usando antes de escribir el código” (E04)
Autorreflexión	R2	Reconocimiento explícito de error conceptual	6/17	“En clase entendí varios de estos elementos, pero a la hora de sentarme a hacerlo me enredé un poco” (E09)
Autorreflexión	R3	Discrepancia expectativa-resultado con carga afectiva	4/17	“Fue mucho más caótico de lo que creí en un inicio” (E01)
Autorreflexión	R4	Cierre sin elaboración	7/17	“El resultado fue el esperado.” (E17, respuesta idéntica en sus 3 ejercicios)
Autorreflexión	R5	Reconocimiento explícito de dependencia de andamiaje externo	2/17	“Si me dijeran que no puedo ver mis otros trabajos como guía, yo creo que no lo sacaría” (E07)

Nota. n = número de estudiantes en cuyas reflexiones se observó la categoría al menos una vez, sobre el total de entregas de cada actividad (Métodos en Arreglos, N=17; Sistema Generativo, N=14). Codificación de un solo analista. Categorías no mutuamente excluyentes.

En la fase de *previsión*, la mayoría de los estudiantes (15 de 17) justificó su elección de método mediante un reconocimiento funcional explícito, sin remitirse únicamente a la

instrucción docente. La mención de inteligencia artificial como estrategia (2 de 17) es un hallazgo que no se anticipaba por ningún instrumento cuantitativo de este estudio. En Sistema Generativo, la mayoría de estudiantes (10 de 14) lograron conectar explícitamente alguna característica de la obra de referencia con una decisión técnica específica, y 4 de 14 explicaron el equilibrio entre control y azar como una decisión de diseño que pasó por una fase de reflexión, el caso más destacado muestra el tipo de activación metacognitiva que en la sección 3.2.5 se plantea como hipótesis frente al mecanismo mediador (Tabla 11, categoría P5).

En la fase de *ejecución*, la tendencia dominante de monitoreo (10 de 17) fue la verificación mediante `console.log()`, lo que claramente es evidencia del ciclo de monitoreo y control que plantean Nelson y Narens (1990, sección 3.2.4). Por otro lado, un grupo más reducido (3 de 17) manifestó autocorrección explícita: el estudiante identifica que una estrategia inicial no funciona y cambia de método antes de terminar (Tabla 11, categoría E2), siendo este el comportamiento que más se aproxima a la autorregulación metacognitiva que mide la subescala MR del MSLQ.

La fase de *autorreflexión* mostró la mayor variación del conjunto: un subgrupo (7 de 17) produjo autoevaluaciones elaboradas con explicaciones claras sobre posibles aplicaciones de los hallazgos a futuro, y un subgrupo de tamaño comparable (7 de 17) cerró la reflexión sin elaboración, en ocasiones con la misma fórmula repetida de forma idéntica en los tres ejercicios. Estos dos patrones no constituyen rasgos fijos: en varios casos coexisten en una

misma persona según el ejercicio, lo que encaja de manera precisa con la distinción entre medición de rasgo y de estado desarrollada en el Capítulo 7 a propósito del MSLQ general en comparación a las encuestas de módulo.

Un hallazgo importante para interpretar el descenso en MR (sección 6.4) es el reconocimiento por parte de los estudiantes de la necesidad de andamiaje externo (2 de 17, categoría R5): “si lo volviera a hacer y me dijeran que no puedo ver mis otros trabajos como guía, yo creo que no lo sacaría porque siempre me falta cosas o ortografía” (E07). Esta clase de declaración, junto con el reconocimiento explícito de bloqueos en otros dos estudiantes (“suelo complicarme pensando que quizá la manera más sencilla no es la efectiva”, E03), sugiere que el descenso en MR no refleja necesariamente una pérdida de estrategias de autorregulación sino una autoevaluación más exigente.

En conjunto, estos juicios de los participantes muestran que los tres momentos del ciclo de Zimmerman (2000) estuvieron presentes, de manera notable en el monitoreo durante la ejecución y mayor heterogeneidad en la autorreflexión. Estos datos, sin embargo, no permiten establecer relaciones estadísticas con el rendimiento o la motivación.

6.6 Prácticas artísticas: módulo de Arte Generativo (OE3)

Se analizó la encuesta de Arte Generativo, aplicada en tres momentos a lo largo del módulo, antes (pre), durante y después (post), sobre 10 ejes temáticos en escala Likert 1–10 (dificultad técnica, reglas generativas, propiedades del objeto, expresión del principio

generativo, planificación, comprensión técnica previa, relación con los pioneros del arte generativo, uso de recursos, gestión del tiempo y descomposición en subtarear). La muestra corresponde a los $N = 13$ estudiantes con MSLQ pre-post completo.

Descriptivos. La Tabla 12 resume la media, mediana y desviación estándar de cada eje en los tres momentos.

Tabla 12

Descriptivos de los 10 ejes de Arte Generativo por momento (escala 1–10, $N = 13$)

Eje	M pre	M dur	M post	DE pre	DE dur	DE post
Q01 — Dificultad técnica	6.85	5.38	6.12	1.86	2.53	1.96
Q02 — Dificultad reglas generativas	6.08	5.77	5.36	2.02	2.13	1.51
Q03 — Propiedades del objeto en pantalla	5.92	5.98	6.97	2.14	1.79	1.42
Q04 — Expresión del principio generativo	5.62	5.54	6.62	2.14	2.11	1.43
Q05 — Plan de acción	4.54	5.06	5.13	1.90	2.47	1.49
Q06 — Comprensión técnica previa	6.23	6.69	5.93	2.20	2.18	2.24
Q07 — Relación con pioneros del arte generativo	5.85	4.99	5.65	1.82	2.69	2.40

Q08 — Uso de recursos	6.31	5.63	5.19	1.60	2.23	2.07
Q09 — Gestión del tiempo	4.62	5.57	6.12	1.76	1.88	1.63
Q10 — Descomposición en subtareas	5.08	5.72	6.29	2.02	2.18	1.98

Comparación de los tres momentos mediante la prueba de Friedman. Dado que se trata de tres mediciones relacionadas (no dos, como en el MSLQ), se utilizó la prueba de Friedman, equivalente no paramétrico del ANOVA de medidas repetidas, para evaluar si existía algún cambio significativo entre pre, durante y post en cada eje. El tamaño del efecto se estimó mediante la *W* de Kendall.

Tabla 13
Prueba de Friedman por eje (N = 13, gl = 2)

Eje	χ^2	p	W de Kendall
Q04 — Expresión del principio generativo	6.049	.049	.233
Q06 — Comprensión técnica previa	5.489	.064	.211
Q03 — Propiedades del objeto en pantalla	5.150	.076	.198
Q09 — Gestión del tiempo	2.652	.266	.102
Q08 — Uso de recursos	2.364	.307	.091
Q05 — Plan de acción	1.674	.433	.064
Q02 — Dificultad reglas generativas	1.565	.457	.060

Q10 — Descomposición en subtareas	1.244	.537	.048
Q07 — Relación con pioneros del arte generativo	1.143	.565	.044
Q01 — Dificultad técnica	1.024	.599	.039

Únicamente el eje Q04 (expresión del principio generativo) mostró un cambio global significativo a través de los tres momentos, $\chi^2(2) = 6.05$, $p = .049$, con un tamaño de efecto pequeño-moderado (W de Kendall = .233). Los ejes Q06 (comprensión técnica previa) y Q03 (propiedades del objeto) se ubicaron en un nivel marginal ($p = .064$ y $.076$, respectivamente), sin alcanzar significación convencional.

Comparaciones post-hoc (solo para Q04). Al ser el único eje con Friedman significativo, se realizaron las tres comparaciones pareadas posibles mediante Wilcoxon, ajustando el umbral de significación por comparaciones múltiples (corrección de Bonferroni: α ajustado = $.05/3 = .017$).

Tabla 14
Comparaciones pareadas post-hoc para Q04 (N = 13)

Comparación	Z	p	r	¿Significativo tras corrección?
Pre vs. Durante	-0.80	.427	.22	No
Durante vs. Post	-1.74	.081	.48	No
Pre vs. Post	-1.73	.084	.48	No

Ninguna de las tres comparaciones pareadas sobrevivió el umbral ajustado, pese a que la prueba global de Friedman había detectado un cambio significativo. El patrón es esperable con un tamaño muestral pequeño ($N = 13$): la señal de cambio existe de forma agregada, pero no es lo bastante fuerte en ninguna transición específica para resistir la corrección por comparaciones múltiples. Las comparaciones durante-post y pre-post muestran los tamaños de efecto más grandes ($r = .48$ en ambas), lo que sugiere que el cambio más relevante ocurre después del momento "durante": la capacidad autopercebida de expresar el principio generativo en el código mejora sobre todo al finalizar la obra y no durante su construcción activa.

6.7 Arte Generativo y logro de aprendizaje

Para explorar la segunda parte de OE3, esto es, la relación entre las prácticas artísticas y el desarrollo de habilidades de programación Frontend, se calculó la correlación de Spearman entre la percepción de cada eje al finalizar el módulo (momento *post*), más un promedio compuesto de los 10 ejes, y el puntaje total del post-test integrado (*Puntaje_bruto*), sobre los mismos $N = 13$ estudiantes.

Tabla 15*Correlación de Spearman entre ejes de Arte Generativo (post) y logro real (N = 13)*

Eje	ρ	p
Promedio compuesto (10 ejes)	.43	.148
Q04 — Expresión del principio generativo	.41	.161
Q05 — Plan de acción	.34	.255
Q07 — Relación con pioneros del arte generativo	.20	.505
Q01 — Dificultad técnica	.18	.567
Q09 — Gestión del tiempo	.12	.694
Q10 — Descomposición en subtarear	-.07	.822
Q06 — Comprensión técnica previa	.06	.844
Q02 — Dificultad reglas generativas	.06	.850
Q08 — Uso de recursos	-.03	.918
Q03 — Propiedades del objeto en pantalla	.01	.963

Ninguna correlación alcanzó significación estadística. La más alta corresponde al promedio compuesto de los 10 ejes ($\rho = .43$, $p = .148$) y a Q04 ($\rho = .41$, $p = .161$), que es el mismo eje con cambio significativo en la Tabla 13. La dirección resulta consistente (quienes sienten haber logrado expresar mejor el principio generativo tienden, aunque sin significación, a obtener mejor puntaje real), pero no permite afirmar una relación estadísticamente sólida entre la experiencia en el módulo de Arte Generativo y el logro técnico general medido por el post-test integrado. Con $N = 13$ y 11 comparaciones sin corrección por comparaciones

múltiples, la potencia estadística disponible es baja; estos resultados deben leerse como exploratorios.

6.8 Prácticas artísticas: módulo de Puntillismo (OE3)

Se analizó la encuesta de Puntillismo, aplicada en pre, durante y post sobre 7 ejes temáticos (escala Likert 1–5), disponible para $N = 12$ estudiantes (de los 13 con MSLQ pre-post completo, uno no presentó esta encuesta). Dentro de esos 12, un estudiante carece de datos en pre y en post, y otro carece de datos en post. En ninguno de esos casos se forzó una recuperación de datos: cada prueba se calculó únicamente con los casos que tenían información completa en las variables requeridas (N varía por prueba).

A diferencia del módulo de Arte Generativo, la revisión de las preguntas mostró que en Puntillismo **no todos los ejes miden el mismo constructo en los tres momentos**: por ejemplo, en el Eje 2 la medición "durante" pregunta por la *satisfacción* con las decisiones artísticas tomadas, mientras que pre y post preguntan por la *dificultad* percibida, de modo que se trata de constructos distintos y no de la misma variable medida en el tiempo. Por esta razón, el tratamiento estadístico se diferenció de forma explícita:

Eje 3 (Comprensión técnica) es el único eje cuyas tres mediciones resultan genuinamente comparables (comprensión autopercebida, misma dirección de escala en los tres momentos).

Se aplicó la prueba de Friedman:

Tabla 16*Eje 3 — Comprensión técnica: descriptivos y prueba de Friedman (N = 10)*

Momento	M	Mdn	DE
Pre	2.30	2.0	0.67
Durante	3.80	4.0	0.79
Post	3.60	3.5	1.17

$\chi^2(2) = 13.27$, $p = .0013$, W de Kendall = .664 (efecto grande). A diferencia de todo lo observado hasta ahora en este capítulo (MSLQ, Arte Generativo), este es un cambio claramente significativo y de magnitud considerable.

Las comparaciones post-hoc (Wilcoxon, corrección de Bonferroni, α ajustado = .017) muestran un patrón preciso:

Tabla 17*Comparaciones pareadas post-hoc para el Eje 3 — Comprensión técnica del módulo de Puntillismo (N = 10)*

Comparación	Z	p	r	¿Significativo tras corrección?
Pre vs. Durante	-2.71	.0067	.86	Sí
Durante vs. Post	-0.63	.527	.20	No
Pre vs. Post	-2.59	.0097	.82	Sí

La comprensión sobre la parte técnica aumenta de forma marcada y significativa entre el pre y el durante (efecto muy grande, $r = .86$), se mantiene estable entre el durante y el post (sin

cambio adicional significativo), y el nivel alcanzado al final sigue siendo significativamente superior al de partida ($r = .82$). Estos datos apuntan a que la comprensión se consolida temprano, justo en el momento de construcción de la obra, y se sostiene hasta el cierre del ejercicio.

Ejes 4 y 6 (2 momentos comparables: pre-post).

Tabla 18

Ejes 4 y 6 del módulo de Puntillismo: prueba de Wilcoxon pre-post ($N = 10$)

Eje	N	M pre	M post	Z	p	r
Eje 4 — Aplicación puntillismo	10	2.60	3.70	-2.33	.020	.74
Eje 6 — Relación arte- programación	10	2.90	3.80	-2.07	.039	.66

Ambos ejes muestran un aumento estadísticamente significativo entre pre y post, con tamaños de efecto grandes. Los estudiantes terminaron el módulo sintiéndose con mayor capacidad de aplicar los principios del puntillismo en su código (Eje 4), y valorando de mejor manera la conexión entre la práctica artística y la programación (Eje 6) de lo que preveían al comienzo.

Ejes 1, 2, 5, 7 e ítems únicos: solo descriptivos, sin prueba de cambio.

Estos ejes agrupan distintas dimensiones entre momentos (ver arriba) o solo se midieron una

vez, por lo que se reportan de forma descriptiva, sin tratarlos como si fueran una serie temporal de la misma variable.

Tabla 19

Descriptivos de ejes e ítems no sometidos a prueba de cambio

Variable	N	M	Mdn	DE
Eje 1 — Dificultad esperada (pre)	11	3.73	4.0	0.65
Eje 1 — Dificultad vs. esperado (post)	10	2.30	2.0	1.16
Eje 2 — Dificultad esperada (pre)	11	2.64	3.0	0.81
Eje 2 — Satisfacción con decisiones (durante)	12	3.83	4.0	1.19
Eje 2 — Dificultad vs. esperado (post)	10	2.10	2.0	1.20
Eje 5 — Claridad del plan (pre)	11	2.27	2.0	1.01
Eje 5 — Desviación del plan (durante)	12	2.92	3.0	0.67
Eje 5 — Deseo de cambiar de estrategia (durante)	12	2.42	2.0	1.00
Eje 5 — Utilidad de las estrategias (post)	10	4.10	4.0	0.74
Eje 7 — Autoeficacia con loops (pre, único)	11	2.27	2.0	0.79

momento)				
Confusión/bloqueo (durante, único momento; escala invertida: 5=nunca)	12	2.92	3.0	0.79
Uso de debugging (durante, único momento)	12	3.08	3.0	1.00
Pausas metacognitivas (durante, único momento)	12	3.75	4.0	0.45
Aprendizaje sobre la propia forma de aprender (post, único momento)	10	3.80	4.0	1.23

En ambos ejes que preguntan "qué tan difícil fue comparado con lo esperado" (Ejes 1 y 2, post), el promedio esta por debajo del punto medio de la escala (3 = exactamente como esperaba), lo que sugiere que los estudiantes encontraron la actividad **más fácil de lo que anticipaban**, tanto en lo técnico como en lo artístico.

La claridad del plan inicial fue baja (Eje 5 pre, M = 2.27), pero la utilidad percibida de las estrategias usadas al final fue alta (Eje 5 post, M = 4.10), lo que constituye una posible señal de que la planificación emergente durante el proceso resultó más relevante que la previa. Las pausas metacognitivas autorreportadas durante el ejercicio (M = 3.75) fueron relativamente frecuentes.

Capítulo 7. Discusión y conclusiones

Este capítulo constituye el cierre del estudio que se realizó, en esta sección se interpreta los resultados que se presentaron en el Capítulo 6 a la luz del marco teórico y de los antecedentes, se revisa qué ocurrió con cada hipótesis, ordena las conclusiones por objetivo específico, admite los límites del trabajo y señala algunas de las rutas que se pueden explorar para futuras investigaciones.

El balance general es de apoyo parcial al modelo explicativo, encontrando que existe un contraste que atraviesa todo el capítulo y conviene tenerlo presente desde el principio.

Cuando la motivación, la autoeficacia y la metacognición se midieron a lo largo del semestre completo con el MSLQ aplicado al inicio y al final, no apareció ningún cambio positivo significativo, de hecho, la autorregulación metacognitiva registró una disminución. Sin embargo, cuando esas mismas dimensiones son medidas de forma secuencial alrededor de una práctica artística concreta mediante encuestas específicas del momento abordado (Arte Generativo y sobre todo las de Puntillismo), se observaron cambios positivos y de magnitud considerable especialmente en los campos de comprensión técnica y confianza para aplicar lo aprendido, de manera autopercebida por parte de los estudiantes.

A eso se suma que se logró identificar una asociación de varios indicadores de autorreporte del post-test, en particular elaboración, organización y ganancia percibida, con el desempeño técnico real. A partir de estas observaciones, se propone como uno de los hallazgos más

relevantes: es posible detectar efectos positivos de las estrategias implementadas a nivel de módulo, sin embargo, estos efectos se disuelven al hacer una comparación en la medición de semestre completo.

7.1 Discusión de resultados frente a la teoría

(a) Autoeficacia (Bandura, 1997). La autoeficacia no registró un cambio significativo entre la medición inicial y la final ($t(12) = 0.49$, $p = .630$, $dz = 0.14$). Con base en los resultados, esta diferencia entre el antes y el después es tan pequeña que bien podría deberse al azar. El dato no respalda la expectativa que se tuvo en el momento inicial, según la cual las experiencias de logro o maestría (estructuradas por la metodología artística), que Bandura considera la fuente más potente de autoeficacia, bastarían para tener efectos significativos en el lapso de un semestre.

Sin embargo, hay un detalle que llama la atención, cuando se compara la autoeficacia con el resultado del logro (sección 6.4), AE se ubicó en un nivel muy cercano al puntaje del post-test ($p = .53$, $p = .062$): los estudiantes con mayor autoeficacia percibida tendieron a conseguir mejores resultados, pero sin significación estadística. Ese vínculo es consistente con la tesis de Bandura, según la cual autoeficacia y desempeño van de la mano. Lo que este estudio no logra demostrar es que la intervención haya incrementado la primera.

(b) Teoría de la Autodeterminación (Ryan y Deci, 2000). La orientación hacia metas intrínsecas tampoco mostró cambios significativos ($t(12) = -0.55$, $p = .596$). Al menos en lo

que capta el MSLQ, no hay certezas para afirmar que la satisfacción de las tres necesidades que postula esta teoría (autonomía, competencia y relación) se refleje en un incremento significativo de la motivación intrínseca.

Vale la pena señalar un límite del instrumento implementado en el estudio. La teoría anticipa efectos sobre la calidad de la motivación y sobre la persistencia frente a la dificultad, dimensiones que el MSLQ-IM no captura directamente: se concentra en el tipo de meta que persigue el estudiante. El análisis cualitativo de las bitácoras (sección 6.5) brindan una evidencia indirecta; al menos dos estudiantes anticipan que una tarea “puede llegar a ser complicado” antes de intentarla, un matiz de calidad motivacional que el cuestionario MSLQ-IM no captura.

(c) Modelo de dos niveles de Nelson y Narens (1990). El hallazgo teóricamente con mayor relevancia del estudio es el descenso significativo en Autorregulación Metacognitiva (MR), $t(12) = 2.22$, $p = .046$, $dz = 0.62$; Wilcoxon $r = .53$. Este resultado va en contra de lo que anticipaba el modelo explicativo, que esperaba justo lo contrario, sin embargo permite una lectura coherente con la literatura sobre calibración metacognitiva; es común que los aprendices novatos sobreestimen su capacidad de monitoreo y control antes de enfrentar la complejidad real de un campo, y es luego de la experiencia directa, que los aprendices desarrollan un criterio mas exigente y realista sobre lo que significa monitorear su propio aprendizaje. En el presente estudio, la experiencia directa consistió en programar viendo el

resultado en pantalla y enfrentar dificultades concretas, como la Sección 3 del post-test, donde solo el 47.1% acertó.

Bajo esa lectura, la disminución en MR no sería evidencia concluyente de que los estudiantes hayan perdido capacidad de autorregularse, sino más bien de que hubo una recalibración metacognitiva. El ciclo de monitoreo que describen Nelson y Narens se activó; lo que muestra el autorreporte es una autoevaluación más coherente, no un puntaje más alto. Las bitácoras respaldan en parte esta lectura, dos de los diecisiete estudiantes mencionan que su desempeño depende de poder consultar trabajos anteriores. Otros dos limitan su propia capacidad de razonamiento. Esto señala un proceso de calibración más exigente en lugar de una pérdida de estrategias.

Dos casos entre diecisiete son pocos, adicionalmente, los datos provienen de bitácoras de actividades del curso y no de los campos del post-test. Por ello, la recalibración metacognitiva queda, entonces, como una interpretación con apoyo parcial, no como algo verificado.

(d) Antecedentes empíricos de los cinco clústeres. El análisis de los antecedentes empíricos de los cinco clústeres revela que Elaboración (EL, $\rho = .70$, $p = .008$) y Organización (OR, $\rho = .63$, $p = .021$), y no la autorregulación metacognitiva (MR), son las subescalas más asociadas al logro real. Este resultado complementa el Clúster B (Loksa et al., 2016; Prather et al., 2019), que reporta beneficios de la orientación metacognitiva

explícita en programación sin diferenciar entre subprocesos específicos de autorregulación. Además aporta una distinción comprobada: dentro del campo de la metacognición y las estrategias de aprendizaje, los procesos de organización y elaboración de la información, y no la autorregulación metacognitiva en sentido estricto, muestran la asociación más consistente con el desempeño técnico.

El antecedente más comparable es el de Cheng, Poon, Lau y Zhou (2019), quienes aplicaron el mismo instrumento (MSLQ) a 66 estudiantes de un curso introductorio de programación universitaria, ellos encontraron que las estrategias cognitivas de orden superior, el control metacognitivo, la gestión del tiempo y la búsqueda de ayuda correlacionaban positivamente con el éxito académico. Los resultados de la presente investigación en que las estrategias de aprendizaje autorregulado predicen el logro en programación, aunque con una diferencia, En Cheng et al., el control metacognitivo sí alcanzó asociación significativa, mientras que en este estudio MR no la alcanzó ($p = .47$, $p = .104$). Dada la coincidencia direccional y de magnitud, la diferencia de potencia estadística ($N = 66$ frente a $N = 13$) es la explicación más indicada. Finalmente, el uso espontáneo inteligencia artificial para resolver problemas (sección 6.5) encuentra un marco interpretativo reciente en Xu, Qiao, Cheng, Liu y Zhao (2025), quienes demostraron experimentalmente que, en entornos de IA generativa, el soporte metacognitivo no interfiere en el logro, pero sí protege la autorregulación, actuando como estrategia para que los estudiantes no sientan que pierden el control de su estudio. Este antecedente ofrece una explicación complementaria, no excluyente de la recalibración

propuesta en 7.1(c), para el descenso observado en MR en un grupo en el que el recurso de asistentes y uso de IA estaba disponible y fue efectivamente utilizado por al menos parte de los estudiantes.

En la parte motivacional (el Clúster C), la relación entre qué tan capaz se siente el estudiante y sus notas rozó la significación estadística ($p = .53$, $p = .062$). Es direccionalmente alineada con Lishinski et al. (2016), aunque de menor magnitud que la reportada en su estudio, lo cual resulta esperable dado el tamaño de participantes considerablemente menor de esta investigación. Aparece una contradicción entre el resultado nulo del MSLQ pre-post y los efectos grandes a nivel de módulo (secciones 6.6–6.8) al comparar con los Clústeres A y D. Conradty y Bogner (2019), en un diseño pre-post-seguimiento con 125 estudiantes, encontraron que un módulo STEAM único produce conocimiento duradero y motivación intrínseca estable, pero no modifica las autoevaluaciones sobre creatividad; su conclusión es que una intervención aislada no basta para cambiar las medidas de ese tipo y que se requieren prácticas constantes. El comportamiento es similar al de esta tesis, con H1 nula en el agregado y efectos detectables cerca de la práctica, y respalda con evidencia la lectura de granularidad rasgo/estado desarrollada en 7.1(e). Por un camino similar, Smit et al. (2024) sí registraron aumentos de autoeficacia con tareas creativas de programación visual, aunque midieron el indicador en proximidad temporal a la tarea y con una muestra veinte veces mayor ($N = 269$): dos condiciones que no estuvieron presentes en la medición pre-post de esta investigación.

Por último, el Clúster A cuenta ahora con un antecedente centrado en la misma herramienta pedagógica de esta tesis. Graßl, Musliu y Fraser (2025) evaluaron un curso escolar de programación creativa basado en arte generativo ($N = 74$) y observaron mejoras de aprendizaje en ambos géneros, sumado a una reducción significativa de estereotipos sobre la programación en las estudiantes, lo que se relaciona con los cambios actitudinales y de comprensión autopercebida registrados aquí alrededor de los módulos artísticos. Ningún hallazgo de este estudio desmiente, además la síntesis del Clúster E sobre la dificultad de los cursos introductorios de programación: que la Sección 3 (comprensión de ciclos for) sea el punto de mayor dificultad coincide con la literatura sobre carga cognitiva en las primeras estructuras de control iterativas.

(e) Granularidad en la medición: un hallazgo emergente. Los resultados de las secciones 6.6–6.8 revelan una característica que no se había contemplado de manera inicial (Capítulo 3). La autoeficacia (MSLQ-AE) no mostró cambios significativos entre el inicio y el final del semestre; ahora bien, la confianza aplicada, medida específicamente en torno al módulo de Puntillismo, sí aumentó significativamente y con un efecto grande (Eje 4, $r = .74$) entre el pre y el post de ese módulo. De manera similar sucede en la comprensión técnica (Eje 3, $r = .82$ pre-post) y en la valoración de la relación arte-programación (Eje 6, $r = .66$). Parece entonces, que las prácticas artísticas generan experiencias de maestría (Bandura, 1997) y activación metacognitiva, pero su efecto es más evidente cuando se mide en proximidad a la

práctica específica, en lugar de medirlo como un cambio agregado y estable en el MSLQ al inicio y al final del semestre. Esta distinción (razgo vs estado) no estaba contemplada en el diseño original y constituye un hallazgo que requiere investigación sistemática en trabajos futuros (sección 7.6).

7.2 Verificación de hipótesis

H1: No soportada a nivel general; matizada por evidencia a nivel de módulo. La hipótesis de un cambio positivo y significativo en motivación intrínseca y autoeficacia, medidas mediante el cuestionario MSLQ no se sostiene con los datos pre-post: ni IM ($t(12) = -0.55, p = .596$) ni AE ($t(12) = 0.49, p = .630$) mostraron cambios significativos, y los tamaños de efecto ($dz = -0.15$ y 0.14 , respectivamente) son mínimos según los criterios convencionales de Cohen. Sin embargo, cuando se mide de forma específica alrededor de una práctica artística puntual, por ejemplo, en las encuestas de módulo, sí se observan incrementos significativos y de efecto grande: Puntillismo Eje 4 (aplicación de principios, $r = .74$) y Eje 6 (valoración de la relación arte-programación, $r = .66$), ambos $p < .05$ (secciones 6.6-6.8). H1 como fue formulada sobre el MSLQ, no se valida, pero sí se matiza su interpretación: la metodología parece producir efectos motivacionales reales y detectables a nivel de módulo específico, que no se traducen en un cambio agregado, estable y medible (ver discusión de granularidad de la medición, sección 7.1e).

H2: No verificable de forma concluyente; evidencia descriptiva favorable. El diseño no permite una verificación del logro de aprendizaje antes vs después, dado que los datos de la

evaluación técnica inicial se perdieron por un error técnico de la plataforma (secciones 4.4.2 y 4.5), una limitación reconocida desde la metodología. Con los datos disponibles, el rendimiento descriptivo en el post-test integrado es aceptable a bueno ($M = 10.59/13$ puntos, 81.5%; $Nota = 4.07/5$), señala un punto de dificultad (Sección 3, 47.1%). Además, que la ganancia percibida por los estudiantes correlacione con la nota real ($\rho = .58, p = .014$) ofrece un buen sintoma de validez concurrente del instrumento, sin embargo, no constituye evidencia de mejora causada por la intervención. Los proyectos entregables y las rúbricas de desempeño previstos como fuentes complementarias de esta hipótesis (sección 4.4.3) han sido analizados de manera superficial en este corte y no constituyen un dato con aportes eficaces. H2 se considera, por tanto, **parcialmente evaluada, con evidencia descriptiva favorable pero sin verificación causal**.

H3: no se cumple en su forma original y requiere una revisión. La hipótesis predecía un aumento en los indicadores de autorregulación metacognitiva en la misma dirección con los cambios en la motivación y el logro de aprendizaje. En cambio, la dirección es opuesta (la MR disminuye en lugar de aumentar) y se observa una falta de asociación significativa con el logro (la MR no es la subescala más predictiva; $\rho = .47, p = .104$, no significativo).

No obstante, la hipótesis no se descarta por completo ya que otras variables relacionadas con la metacognición y la autorregulación del aprendizaje como son la elaboración, la organización y las creencias sobre el control del aprendizaje, sí muestran una asociación significativa con el logro. Esto sugiere que el mecanismo de mediación podría estar

presente, pero a través de un conjunto más amplio de estrategias de aprendizaje, no solamente la subescala MR del MSLQ

7.3 Conclusiones por objetivo específico

OE1 (motivación pre-post). Se caracterizó el nivel de motivación de los estudiantes en las 10 subescalas del MSLQ antes y después de la intervención ($N = 13$ pares completos). La conclusión es que, en el corte actual de datos, la metodología no modificó de forma estadísticamente significativa los niveles de motivación intrínseca/extrínseca, valor de la tarea, creencias de control, autoeficacia o ansiedad; el único cambio significativo detectado fue el descenso en la autorregulación metacognitiva, que corresponde conceptualmente más a la variable mediadora (OE4) que a la motivación en sentido estricto.

OE2 (logro de aprendizaje). Para medir el aprendizaje se aplicó un instrumento integrado de post-test ($N = 17$), el resultado promedio fue de 81.5% (Nota = 4.07/5) con alta variabilidad entre secciones temáticas (47.1%–100%). El recorrido que partió de p5.js → JavaScript puro muestra haber funcionado en tipos de datos, arrays y DOM, también se observa una debilidad marcada en la comprensión de los ciclos for (Sección 3). La conclusión es, no obstante, provisional: los datos iniciales de conocimiento se perdieron y los proyectos entregables por los estudiantes están pendientes de un análisis en profundidad.

OE3 (documentación de prácticas artísticas). Este objetivo se completó para los dos módulos con datos disponibles (Arte Generativo y Puntillismo; los módulos de Pensamiento

Algorítmico y Abstraccionismo no cuentan con ningún instrumento de encuesta similar en este corte).

En Arte Generativo, de los 10 ejes temáticos evaluados en tres momentos, únicamente la expresión del principio generativo (Q04) mostró un cambio global significativo (Friedman, $\chi^2(2) = 6.05$, $p = .049$), sin que ninguna comparación sobreviviera a la corrección por comparaciones múltiples; su la relación con el logro técnico real no alcanzó significancia estadística ($p = .43$, $p = .148$).

Para Puntillismo, el panorama es relevante: la comprensión técnica que los estudiantes autoreportan (Eje 3) aumentó de forma significativa y con efecto grande entre pre y durante ($r = .86$) y se mantuvo alta hasta el post (pre vs. post, $r = .82$); la aplicación percibida de los principios del puntillismo (Eje 4) y la valoración de la relación arte-programación (Eje 6) también aumentaron significativamente entre pre y post ($r = .74$ y $.66$, respectivamente). En conjunto los hallazgos documentan que las prácticas artísticas sí se relacionan con cambios autopercebidos relevantes en la comprensión técnica y la confianza aplicada aunque la evidencia de una relación con el logro (medido por el post-test integrado) sigue siendo débil.

OE4 (metacognición como mediadora). Se analizó la relación entre los procesos metacognitivos activados durante la metodología y los cambios en motivación y logro. La conclusión requiere una revisión ssituada: la autorregulación metacognitiva medida por el MSLQ (MR), disminuyó significativamente tras la intervención y no constituye la subescala

más asociada al logro real. En cambio, la elaboración y la organización de la información, procesos cognitivos, muestran una relación significativa.

OE5 (voz del estudiante). Este objetivo **se alcanzó parcialmente**. El análisis de contenido temático de las bitácoras de aprendizaje de tres actividades del curso (sección 6.5; N = 17, 13 y 14 según la actividad) documenta los tres momentos del ciclo de autorregulación de Zimmerman (2000), con especial consistencia en la fase de ejecución (monitoreo mediante verificación incremental, 10 de 17 estudiantes) y mayor heterogeneidad en la autorreflexión. La evidencia disponible respalda tentativamente, aunque de manera no concluyente, la interpretación de "recalibración metacognitiva" propuesta para el descenso en MR (OE1/OE4, sección 7.1c).

7.4 Limitaciones del estudio

Las limitaciones que corresponden al diseño metodológico comprenden:

En primer lugar, la ausencia de un grupo control, lo que impide atribuir causalidad con certeza y limita la interpretación a asociaciones entre la intervención y los cambios observados.

En segundo lugar, el sesgo de rol dual del investigador, quien actúa simultáneamente como docente e investigador, lo que puede afectar tanto la implementación de la metodología como la recolección de datos cualitativos.

En tercer lugar, la utilización del then-test retrospectivo como estimación de los saberes previos, solución adoptada ya que se perdieron los datos de la evaluación diagnóstica inicial

por un error técnico de la plataforma (sección 4.4.2), lo que genera una medida de aquello que los estudiantes creen recordar que sabían en lugar de una medida objetiva del estado inicial.

cuarto, el tamaño de muestra limitado, propio de una investigación con el grupo natural del aula, lo que restringe la generalización de los resultados.

Quinto, a las limitaciones de diseño se agregan restricciones específicas del análisis estadístico presentado en el Capítulo 6: las correlaciones exploratorias entre subescalas MSLQ y logro (Tabla 10, sección 6.4) se calcularon sin aplicar corrección por comparaciones múltiples y sobre una muestra de únicamente 13 estudiantes. Estos hallazgos deben considerarse, en consecuencia, generadores de hipótesis para investigaciones futuras con mayor potencia estadística, y no resultados confirmatorios.

Sexto, el tamaño muestral efectivo varía según el cruce de instrumentos requerido ($N = 17$ para análisis del post-test únicamente, $N = 13$ para análisis que cruzan MSLQ y logro), de ahí que cada resultado se reporte con su N particular, tal como se hizo en el Capítulo 6.

Septimo, para futuras aplicaciones conviene mantener el mismo constructo subyacente en los tres momentos de medición, a fin de posibilitar un análisis longitudinal completo.

Finalmente, de los 13 estudiantes con MSLQ completo, únicamente 10 disponen de datos completos en las variables necesarias para las pruebas de cambio de Puntillismo (uno no presentó la encuesta en ningún momento y dos presentan momentos faltantes), lo que reduce aún más la limitada potencia estadística disponible para estos análisis específicos.

7.5 Contribuciones de la investigación

Esta investigación realiza cuatro contribuciones originales al campo, independientemente de la verificación parcial de las hipótesis. En primer lugar, propone y somete a prueba empírica un modelo metodológico que articula prácticas artísticas, metacognición y motivación en el dominio específico de la programación Frontend, una intersección identificada como un vacío investigativo no abordado sistemáticamente en la revisión de la literatura (Capítulo 2). En segundo lugar, el hallazgo de que los efectos motivacionales y de comprensión técnica de las prácticas artísticas son detectables de manera óptima a nivel de módulo específico (Puntillismo, con tamaños de efecto grandes, r entre .66 y .86), en contraste con el nivel de semestre (MSLQ), aporta datos de investigación relevante sobre la **granularidad temporal de la medición** en estudios de intervenciones educativas breves.

En tercer lugar, el instrumento integrado de post-test, que combina en una sola aplicación desempeño técnico objetivo, then-test retrospectivo e ítems de autorreporte metacognitivo y motivacional por tema, representa una solución metodológica potencialmente replicable para contextos donde, como en este estudio, no fue posible aplicar una prueba diagnóstica técnica objetiva antes de la intervención, con la limitante mencionada de que puede existir contaminación de ítem (P9) por adivinanza retrospectiva.

En cuarto lugar, el estudio aporta evidencia empírica desde un contexto universitario latinoamericano (Colombia) sobre el efecto de prácticas artísticas en la enseñanza de programación. El contexto explorado cuenta con hace contribuye ante la escasa

representación en la literatura revisada (Capítulo 2, Clúster E), y lo hace con transparencia metodológica respecto a sus propias limitaciones de potencia estadística y diseño.

7.6 Recomendaciones para la implementación de la metodología

Las conclusiones anteriores tienen consecuencias prácticas para quien quiera aplicar esta metodología en un curso introductorio de programación. Las recomendaciones que se presentan a continuación se derivan de lo ocurrido en este curso concreto, e incluyen tanto lo que resulta conveniente conservar como lo que conviene ajustar.

Sobre la secuencia de módulos. El módulo de Puntillismo produjo los cambios más marcados de todo el estudio, con incrementos grandes y significativos en comprensión técnica autopercebida ($r = .86$ entre el pre y el durante) y en confianza aplicada ($r = .74$). Tres rasgos de su diseño explican de forma posible ese resultado y merecen conservarse. El referente artístico permite una traducción visual directa a la estructura del código y el criterio de logro es evidente a simple vista. Los módulos cuyo referente resulta más abstracto no ofrecen esa correspondencia inmediata; conviene, por ello, situarlos después, cuando el estudiante ya cuenta con un desarrollo técnico.

Sobre el punto de dificultad detectado. La comprensión del ciclo for en el contexto p5.js fue el resultado del post-test con menor acierto por parte de los estudiantes, 47.1% de respuestas correctas frente a un promedio general de 81.5%. Un posible ajuste a futuro podría ser mencionar la diferencia entre el bucle implícito de dibujo que contiene p5.js por medio de la función draw y el bucle explícito que se escribe dentro de esa misma función.

Quien replique la secuencia podría tratar esa distinción de forma directa en lugar de asumirla por evidente.

Sobre las estrategias metacognitivas a fortalecer . Las subescalas de elaboración ($\rho = .70$) y organización ($\rho = .63$), junto con las creencias de control del aprendizaje ($\rho = .61$) fueron las que estuvieron en relación con el logro real. Conviene diseñar sesiones para que los estudiantes puedan elaborar y organizar la información técnica, así como realizar pausas de monitoreo.

Sobre cuándo medir. El hallazgo de granularidad expuesto en la sección 7.1(e) tiene una consecuencia operativa, conviene aplicar instrumentos breves inmediatamente antes y después de cada módulo, y reservar el cuestionario extenso para el encuadre general del curso.

Sobre el descenso en el autorreporte metacognitivo. La regulación metacognitiva presentó un valor estadísticamente significativo pero negativo tras la intervención, lo cual no implica necesariamente un deterioro del aprendizaje ni de la estrategia de metacognición observada durante el proceso de investigación. Un investigador que interprete ese descenso como un fracaso podría anular decisiones acertadas, por lo tanto, conviene revisar la literatura o consultar a pares académicos para justificar.

Sobre el uso de asistentes de inteligencia artificial. El análisis de bitácoras documentó el recurso a herramientas de IA como estrategia de resolución, algo que ningún instrumento del estudio había previsto (sección 6.5). Antes que prohibirlo o ignorarlo, conviene incorporarlo

al andamiaje metacognitivo pidiendo al estudiante que declare en qué punto recurrió a la herramienta, qué le solicitó y qué modificó de la respuesta obtenida. Xu et al. (2025) mostraron, en esa línea, que el soporte metacognitivo explícito en entornos de IA generativa protege la autorregulación, aun cuando no modifique el logro.

7.7 Proyecciones y líneas de trabajo futuro

Primera, sumar un grupo de comparación y una medición objetiva de entrada permitiría atribuir causalidad a los cambios observados y con ello resolver la principal limitación de este estudio (H2 no verificable de forma concluyente por la pérdida de la línea base objetiva). **Segunda**, una réplica con una muestra de mayor tamaño es necesaria para dotar de potencia estadística suficiente al análisis mediacional entre las subescalas MSLQ y el logro (sección 6.4): con $N = 13$, hallazgos con tamaños de efecto moderados (MR: $\rho = .47$) no alcanzan significación estadística, y no puede descartarse que una muestra mayor confirme o refute la matización de H3 propuesta en este estudio. **Tercera**, el análisis de contenido temático de las bitácoras de aprendizaje se completó (sección 6.5) y ofrece una primera lectura cualitativa del hallazgo contraintuitivo del descenso en MR; sin embargo, la observación y codificación de otro analista podría plantear una línea de trabajo más directa para confirmar o refutar de forma más sólida la interpretación de recalibración metacognitiva propuesta en la sección 7.1(c). **Cuarta**, una línea de investigación sobre estereotipos del perfil del programador, explorando si la metodología artística modifica las creencias previas de los estudiantes sobre quién puede programar, en línea con los hallazgos

de Graßl et al. (2025) y Bares et al. (2019) reseñados en el Clúster A. **Quinta**, el desarrollo de un Ambiente Virtual de Aprendizaje (AVA) basado en los cuatro módulos artísticos documentados en el Capítulo 5, que permitiría escalar y replicar la metodología más allá del contexto de aula única de este estudio, e integrar instrumentación de recolección de datos más robusta desde el diseño (incluyendo diagnóstico técnico pre-intervención y registro estructurado de bitácoras) para superar las limitaciones documentadas en la sección 7.4. **Sexta**, investigar de forma sistemática la distinción entre medición de rasgo y de estado propuesta en la sección 7.1(e).

Referencias Bibliográficas

ACM/IEEE-CS/AAAI Joint Task Force on Computing Curricula. (2024). *Computer Science Curricula 2023*. ACM Press, IEEE Computer Society Press & AAAI Press.

<https://doi.org/10.1145/3664191>

Aguilera, D., & Ortiz-Revilla, J. (2021). STEM vs. STEAM Education and Student Creativity: A Systematic Literature Review. *Education Sciences*, 11(7), 331.

<https://doi.org/10.3390/EDUCSCI11070331>

Bandura, A. (1997). *Self-efficacy: The exercise of control*. Freeman.

Bares, W. H., Manaris, B. Z., McCauley, R., & Moore, C. (2019). Achieving gender balance through creative expression. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE)*. <https://doi.org/10.1145/3287324.3287435>

Barsalou, L. W. (2008). Grounded cognition. *Annual Review of Psychology*, 59(1), 617–645. <https://doi.org/10.1146/annurev.psych.59.103006.093639>

Bautista, G., Malacapo, J., Gallos-Cronberg, F., Kasti, H., Dana-Picard, N., & Lavicza, Z. (2025). Function Art: Linking Mathematics, Technology, and Visual Arts. *School Science and Mathematics*. <https://doi.org/10.1111/ssm.18373>

Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *ACM SIGCSE Bulletin*, 39(2), 32–36. <https://doi.org/10.1145/1272848.1272879>

Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education: Implications for policy and practice*.

Publications Office of the European Union.

Brown, A. L. (1987). Metacognition, executive control, self-regulation, and other more mysterious mechanisms. En F. E. Weinert & R. H. Kluwe (Eds.), *Metacognition, motivation, and understanding* (pp. 65–116). Erlbaum.

Cheng, G., Poon, L. K. M., Lau, W. W. F., & Zhou, R. C. (2019). Exploring the relationship between self-regulated learning strategies and computer programming achievement in higher education. *Proceedings of the 5th International Conference on Education*, 5, 67–74.

<https://doi.org/10.17501/24246700.2019.5108>

Cheryan, S., Master, A., & Meltzoff, A. N. (2015). Cultural stereotypes as gatekeepers: Increasing girls' interest in computer science and engineering. *Frontiers in Psychology*, 6, 49. <https://doi.org/10.3389/fpsyg.2015.00049>

Conradty, C., & Bogner, F. X. (2019). From STEM to STEAM: Cracking the code? How creativity & motivation interacts with inquiry-based learning. *Creativity Research Journal*, 31(3), 284–295. <https://doi.org/10.1080/10400419.2019.1641678>

Conradty, C., Sotiriou, S. A., & Bogner, F. X. (2020). How creativity in STEAM modules intervenes with self-efficacy and motivation. *Education Sciences*, 10(3), 70.

<https://doi.org/10.3390/educsci10030070>

Corrigan, M. W., Wong, J. T., Grove, D., Andersen, S., & Hughes, B. S. (2025). Enhancing elementary students' conceptual understandings of scientific phenomena: The impact of STEAM-First and STEM-First approaches. *Science Education*, 109(5), 1336–1364.
<https://doi.org/10.1002/sce.21942>

Dweck, C. S. (2000). *Self-theories: Their role in motivation, personality, and development*. Psychology Press.

Elliot, A. J. (1999). Approach and avoidance motivation and achievement goals. *Educational Psychologist*, 34(3), 169–189. https://doi.org/10.1207/s15326985ep3403_3

Flavell, J. H. (1976). Metacognitive aspects of problem solving. En L. B. Resnick (Ed.), *The nature of intelligence* (pp. 231–235). Erlbaum.

Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry. *American Psychologist*, 34(10), 906–911.
<https://doi.org/10.1037/0003-066X.34.10.906>

Fragapane, V., & Standl, B. (2021). Work in progress: Creative coding and computer science education — From approach to concept. *2021 IEEE Global Engineering Education Conference (EDUCON)*. <https://doi.org/10.1109/EDUCON46332.2021.9453951>

Galanter, P. (2003). What is generative art? Complexity theory as a context for art theory. *Proceedings of the 6th Generative Art Conference*, 225–235.

García-Peñalvo, F. J. (2016). What computational thinking is. *Journal of Information Technology Research*, 9(3), v–viii.

Graßl, I., Musliu, X., & Fraser, G. (2025). Programming art: Does creative programming shift girls' stereotypes, affections, and comprehension of programming? *Proceedings of the ACM Global Computing Education Conference (CompEd 2025)*.

<https://doi.org/10.1145/3736181.3747161>

Greenberg, I., Kumar, D., & Xu, D. (2012). Creative coding and visual portfolios for CS1. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE)*, 247–252. <https://doi.org/10.1145/2157136.2157214>

Henriksen, D. (2014). Full STEAM ahead: Creativity in excellent STEM teaching practices. *The STEAM Journal*, 1(2), 1–9. <https://doi.org/10.5642/steam.20140102.15>

Henriksen, D., Mehta, R., & Mehta, S. (2019). Design thinking gives STEAM to teaching: A framework that breaks disciplinary boundaries. En S. M. Khine & S. Areepattamannil (Eds.), *STEAM Education: Theory and Practice*. Springer. <https://doi.org/10.1007/978-3-030-04003-1>

IBM Corp. (2019). *IBM SPSS Statistics for Windows* (Versión 26.0) [Software de computador]. IBM Corp. <https://www.ibm.com/products/spss-statistics>

Jawad, H. M., Tout, S., Abualkibash, M., & Xie, Y. (2018). Integrating art and animation in teaching computer programming for high school students: Experimental study. *2018 IEEE*

International Conference on Electro/Information Technology (EIT).

<https://doi.org/10.1109/EIT.2018.8500178>

Kafai, Y. B., & Burke, Q. (2014). *Connected code: Why children need to learn programming*. MIT Press. <https://doi.org/10.7551/mitpress/9992.001.0001>

Katz, I., Eilott, K., & Nevo, N. (2013). 'I'll do it later': Type of motivation, self-efficacy and homework procrastination. *Motivation and Emotion*, 38(1), 111–119.

<https://doi.org/10.1007/s11031-013-9366-1>

Kirschner, P. A. (2009). Epistemology or pedagogy, that is the question. En S. Tobias & T. M. Duffy (Eds.), *Constructivist instruction: Success or failure?* (pp. 144–157). Routledge.

<https://doi.org/10.4324/9780203878842>

Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist*, 41(2), 75–86.

https://doi.org/10.1207/s15326985ep4102_1

Knochel, A. D., & Patton, R. M. (2015). If art education then critical digital making: Computational thinking and creative code. *Studies in Art Education*, 57(1), 21–38.

<https://doi.org/10.1080/00393541.2015.11666280>

Lishinski, A., Yadav, A., Good, J., & Enbody, R. (2016). Learning to program: Gender differences and interactive effects of students' motivation, goals, and self-efficacy on

performance. *Proceedings of the 2016 ACM Conference on International Computing Education Research (ICER)*. <https://doi.org/10.1145/2960310.2960329>

Loksa, D., & Ko, A. J. (2016). The role of self-regulation in programming problem solving process and success. *Proceedings of the 2016 ACM Conference on International Computing Education Research (ICER)*, 83–91. <https://doi.org/10.1145/2960310.2960334>

Loksa, D., Ko, A. J., Jernigan, W., Oleson, A., Mendez, C. J., & Burnett, M. M. (2016). Programming, problem solving, and self-awareness: Effects of explicit guidance. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 1449–1461. <https://doi.org/10.1145/2858036.2858252>

Loksa, D., Margulieux, L., Becker, B. A., Craig, M., Denny, P., Pettit, R., & Prather, J. (2022). Metacognition and self-regulation in programming education: Theories and exemplars of use. *ACM Transactions on Computing Education*. <https://doi.org/10.1145/3487050>

López-García, A., Urquiza-Fuentes, J., Mendes, A. J., & Caeiro-Rodríguez, M. (2025). Improving university students' learning of programming using customised pseudocode. *Computer Applications in Engineering Education*, 33(4). <https://doi.org/10.1002/cae.70061>

Malik, S. I., Mathew, R., Al-Sideiri, A., Jabbar, J., et al. (2021). Enhancing problem-solving skills of novice programmers in an introductory programming course. *Computer Applications in Engineering Education*, 30(1), 174–194. <https://doi.org/10.1002/cae.22450>

Martins, V. F., Concilio, I. A. S., & Guimarães, M. P. (2018). Problem based learning associated to the development of games for programming teaching. *Computer Applications in Engineering Education*, 26(5), 1577–1589. <https://doi.org/10.1002/cae.21968>

McCarthy, L. (2014). *p5.js: A JavaScript library for creative coding*. Processing Foundation. <https://p5js.org>

Nelson, T. O., & Narens, L. (1990). Metamemory: A theoretical framework and new findings. En G. Bower (Ed.), *The psychology of learning and motivation* (Vol. 26, pp. 125–173). Academic Press. [https://doi.org/10.1016/S0079-7421\(08\)60053-5](https://doi.org/10.1016/S0079-7421(08)60053-5)

O'Connor, C., & Joffe, H. (2020). Intercoder reliability in qualitative research: Debates and practical guidelines. *International Journal of Qualitative Methods*, 19, 1–13. <https://doi.org/10.1177/1609406919899220>

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.

Park, T. H., & Wiedenbeck, S. (2011). Learning web development: Challenges at an earlier stage of computing education. *Proceedings of the Seventh International Workshop on Computing Education Research (ICER)*, 125–132. <https://doi.org/10.1145/2016911.2016937>

Pechorina, Y., Anderson, K., & Denny, P. (2023). Metacodenition: Scaffolding the problem-solving process for novice programmers. *Proceedings of the 25th Australasian Computing Education Conference (ACE '23)*, 59–68. <https://doi.org/10.1145/3576123.3576130>

- Pereira, F. D., Oliveira, E. H. T., Oliveira, D. B. F., Cristea, A. I., Carvalho, L. S. G., Fonseca, S. C., Toda, A., & Isotani, S. (2020). Using learning analytics in the Amazonas: Understanding students' behaviour in introductory programming. *British Journal of Educational Technology*, 51(4), 955–972. <https://doi.org/10.1111/bjet.12953>
- Perignat, E., & Katz-Buonincontro, J. (2019). STEAM in practice and research: An integrative literature review. *Thinking Skills and Creativity*, 31, 31–43. <https://doi.org/10.1016/j.tsc.2018.10.002>
- Pintrich, P. R., Smith, D. A. F., García, T., & McKeachie, W. J. (1991). *A manual for the use of the Motivated Strategies for Learning Questionnaire (MSLQ)*. University of Michigan, National Center for Research to Improve Postsecondary Teaching and Learning. <https://files.eric.ed.gov/fulltext/ED338122.pdf>
- Prather, J., Becker, B. A., Craig, M., Denny, P., Loksa, D., & Margulieux, L. (2020). What do we think we think we are doing? Metacognition and self-regulation in programming. *Proceedings of the 2020 ACM Conference on International Computing Education Research (ICER)*, 2–13. <https://doi.org/10.1145/3372782.3406263>
- Prather, J., Pettit, R., Becker, B. A., Denny, P., Loksa, D., Peters, A., Albrecht, Z., & Masci, K. (2019). First things first: Providing metacognitive scaffolding for interpreting problem prompts. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE)*. <https://doi.org/10.1145/3287324.3287374>

Ramírez-Echeverry, J. J., García Carrillo, À., & Olarte Dussán, F. A. (2016). Adaptation and validation of the Motivated Strategies for Learning Questionnaire —MSLQ— in engineering students in Colombia. *International Journal of Engineering Education*, 32(4), 1774–1787. <https://upcommons.upc.edu/handle/2117/107554>

Ramírez-Echeverry, J. J., Restrepo-Calle, F., & Torres Jiménez, S. (2025). Self-regulated learning strategies in computer programming education. *European Journal of Education*, 60(1). <https://doi.org/10.1111/ejed.70052>

Reas, C., & Fry, B. (2007). *Processing: A programming handbook for visual designers and artists*. MIT Press.

Resnick, M. (2007). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. *Proceedings of the 6th ACM SIGCHI Conference on Creativity & Cognition*, 1–6. <https://doi.org/10.1145/1254960.1254961>

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137–172. <https://doi.org/10.1076/csed.13.2.137.14200>

Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, 55(1), 68–78. <https://doi.org/10.1037/0003-066X.55.1.68>

Smit, R., Schmid, R., & Robin, N. (2024). Experiencing enjoyment in visual programming tasks promotes self-efficacy and reduces the gender gap. *British Journal of Educational Technology*, 56(3), 1231–1247. <https://doi.org/10.1111/bjet.13523>

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing computational thinking to young learners: Practicing computational perspectives through embodiment in mathematics education. *Technology, Knowledge and Learning*, 22(3), 443–463.

<https://doi.org/10.1007/s10758-017-9328-x>

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257–285. https://doi.org/10.1207/s15516709cog1202_4

Tomić, B., Stojanović, T., Antović, I., & Milić, M. (2025). Students' test anxiety and performance in introductory programming: Do exam and assessment modalities play a role? *Computer Applications in Engineering Education*, 33(3). <https://doi.org/10.1002/cae.70026>

Varela, F. J., Thompson, E., & Rosch, E. (1991). *The embodied mind: Cognitive science and human experience*. MIT Press. <https://doi.org/10.7551/mitpress/6730.001.0001>

Veenman, M. V. J., Van Hout-Wolters, B. H. A. M., & Afflerbach, P. (2006). Metacognition and learning: Conceptual and methodological considerations. *Metacognition and Learning*, 1(1), 3–14. <https://doi.org/10.1007/s11409-006-6893-0>

Wilson, M. (2002). Six views of embodied cognition. *Psychonomic Bulletin & Review*, 9(4), 625–636. <https://doi.org/10.3758/BF03196322>

Xu, X., Qiao, L., Cheng, N., Liu, H., & Zhao, W. (2025). Enhancing self-regulated learning and learning experience in generative AI environments: The critical role of metacognitive support. *British Journal of Educational Technology*, 56(5), 1842–1863.

<https://doi.org/10.1111/bjet.13599>

Zhang, W., Guan, Y., Hu, Z., Wei, Y., & Xu, D. (2025). Synergistic approaches in education: Elevating computational thinking and metacognitive skills through combined project-based and pair programming learning in high schools. *Educational Technology Research and Development*. <https://doi.org/10.1007/s11423-025-10555-1>

Zimmerman, B. J. (2000). Attaining self-regulation: A social cognitive perspective. En M. Boekaerts, P. R. Pintrich & M. Zeidner (Eds.), *Handbook of self-regulation* (pp. 13–39). Academic Press.

Anexos

Anexo A — Instrumento MSLQ (55 ítems)

Link Preguntas aplicadas:

<https://docs.google.com/spreadsheets/d/1fhnSmTKPgVn3z3g3GQpsDGcIbDWjHGfQ2L4nLWSWNUk/edit?usp=sharing>

Link Respuestas de participantes:

<https://docs.google.com/spreadsheets/d/17ZvoRpPGSS0Q-EJb1dvvpMzGmXSp1EpoJwHRuPEKBr0/edit?usp=sharing>

Anexo B — Instrumento integrado de post-test

Link Preguntas aplicadas:

<https://docs.google.com/document/d/17srXB6gYsu2RQoWom1fQ73ac5UGMEB-WTeilMRqTYtY/edit?usp=sharing>

Link Respuestas de participantes:

<https://docs.google.com/spreadsheets/d/16Em4qh8Smd2lms5vRA23CEYZt3jyX3My/edit?usp=sharing&ouid=104485867374126467867&rtpof=true&sd=true>

Anexo C — Formato de consentimientos informados

Link:

<https://drive.google.com/drive/folders/1NCgXYvjvTkd14Pce9WXtiKW2eVETe8Q0?usp=sharing>

Anexo D — Encuestas de puntillismo

Link:

<https://drive.google.com/drive/folders/1YT8t5nNXUtNBeFqRHhbZI9I47-y3odi4?usp=sharing>

Anexo E — Encuestas arte generativo (previo, durante, después)

Link:

https://docs.google.com/spreadsheets/d/1UAr9EdyUTZEdESRVgDHraE58gxyOBII2-J0mgye7YXU/edit?usp=drive_link

Anexo F — Presentación proyecto final Arte Generativo

[muestraLenguajesDigitales2.mp4](#)

Anexo G — Muestra de código elaborado por estudiantes

Link:

https://drive.google.com/drive/folders/1d1cfqamCjkEbFwpzeqD_FPW8ZEObz2EP?usp=sharing

Anexo H — Documento de uso de IA y comandos implementados

Link:

https://docs.google.com/document/d/1Ngdx_UIFQE2icbtSgTOKZD6c6HpxC54yEOqSODkh9wU/edit?usp=sharing