

Developing Computational Thinking in STEM Degrees through Gamification Strategies: Methodological Proposal and Preliminary Evidence at the National University of Rafaela

Javier Fornari¹  0009-0004-1701-3526 Mariel Lopez¹ Claudia Deco¹ Luciana Burzacca¹

¹ Universidad Nacional de Rafaela, Argentina

javier.fornari@unraf.edu.ar marielopez@unraf.edu.ar
claudia.deco@unraf.edu.ar luciana.burzacca@unraf.edu.ar

Abstract. Computational thinking has been recognized as a core competency in the training of STEM professionals, integrating skills such as problem decomposition, abstraction, pattern recognition, and algorithm design. However, traditional pedagogical approaches have shown significant limitations in effectively promoting these skills, leading to challenges in student motivation and deep learning. This paper presents a methodological proposal for the explicit integration of computational thinking development in STEM programs at the Universidad Nacional de Rafaela, Argentina, through gamification-based strategies. The proposal is grounded in the self-determination theory framework and draws on a systematic review of ten empirical studies that document the effectiveness of gamified interventions across educational levels. The methodology includes an initial diagnostic assessment of students' computational thinking levels, followed by the design and implementation of interdisciplinary gamified activities using both digital tools (Scratch, Blockly, MIT App Inventor, Code.org) and adapted physical educational games. A mixed-methods evaluation framework combining standardized pre-post assessments, motivation surveys, and qualitative techniques is outlined. Preliminary enrollment and retention data from three STEM programs (n=124 new students across 2023–2025) are presented as contextual evidence motivating the intervention. The contribution includes a detailed didactic sequence organized into four thematic units, a rubric-based assessment scheme aligned with computational thinking sub-competencies, and a discussion of implementation considerations for the regional educational context. Expected outcomes include significant improvements in computational thinking competencies, increased intrinsic motivation, and the generation of validated, transferable didactic resources.

Keywords: computational thinking, gamification, STEM education, educational innovation, didactic sequence, assessment rubrics.

Desarrollo del Pensamiento Computacional en carreras STEM mediante Estrategias de Gamificación: Propuesta Metodológica y Evidencia Preliminar en la Universidad Nacional de Rafaela

Resumen. El pensamiento computacional se reconoce como una competencia central en la formación de profesionales STEM, al integrar habilidades de descomposición de problemas, abstracción, reconocimiento de patrones y diseño de algoritmos. Sin embargo, los enfoques pedagógicos tradicionales han mostrado limitaciones significativas para promover estas habilidades de manera efectiva, generando desafíos en la motivación y el aprendizaje profundo de los estudiantes. Este trabajo presenta una propuesta metodológica para la integración explícita del desarrollo del pensamiento computacional en carreras STEM de la Universidad Nacional de Rafaela, Argentina, mediante estrategias basadas en gamificación. La propuesta se fundamenta en la teoría de la autodeterminación y se apoya en una revisión sistemática de diez estudios empíricos que documentan la efectividad de intervenciones gamificadas en diversos niveles educativos. La metodología contempla un diagnóstico inicial del nivel de pensamiento computacional, seguido del diseño e implementación de actividades interdisciplinarias gamificadas con herramientas digitales y juegos educativos adaptados. Se presenta un esquema de evaluación mixto que combina mediciones pre-post, encuestas de motivación y técnicas cualitativas. Se incluyen datos preliminares de matrícula y retención de tres carreras STEM (n=124 ingresantes en 2023-2025) como evidencia contextual. La contribución incluye una secuencia didáctica detallada, un esquema de evaluación basado en rúbricas y una discusión de consideraciones de implementación para el contexto educativo regional.

Palabras clave: pensamiento computacional, gamificación, educación STEM, innovación educativa, secuencia didáctica, rúbricas de evaluación.

1 Introducción

La formación de profesionales en las disciplinas de ciencias, tecnología, ingeniería y matemáticas (STEM) demanda el desarrollo de competencias que trascienden el dominio de herramientas específicas y se orientan a la capacidad de formular, descomponer y resolver problemas complejos en términos computacionales (Wing, 2006; Denning, 2009). En este contexto, el pensamiento computacional (PC) se ha consolidado como un constructo central que integra habilidades tales como la descomposición de problemas, la abstracción, el reconocimiento de patrones, el diseño de algoritmos y la evaluación de soluciones (Wing, 2006; Shute, Sun & Asbell-Clarke, 2017). Diversos autores han planteado que estas habilidades constituyen un núcleo transversal de la educación en computación y resultan fundamentales para el desempeño profesional en áreas que van desde el desarrollo de software hasta la ingeniería de datos y los sistemas inteligentes (Bocconi et al., 2016; Zhang, Herman & Wang, 2021).

Sin embargo, los métodos educativos tradicionales han mostrado limitaciones significativas para promover efectivamente estas habilidades en estudiantes de carreras STEM, lo que genera desafíos en términos de motivación y aprendizaje profundo (Grover & Pea, 2013). En los cursos iniciales de programación y algoritmos, las habilidades del PC suelen trabajarse de manera implícita, sin que sus componentes se planteen como objetivos de aprendizaje explícitos ni se dispongan de estrategias didácticas sistemáticas para promoverlos (Shute et al., 2017; Bocconi et al., 2016). Esta situación contribuye a dificultades recurrentes en la articulación entre la comprensión de problemas, el modelado de estructuras de datos y la implementación de soluciones eficientes.

Uno de los enfoques pedagógicos que ha demostrado ser particularmente eficaz para desarrollar el PC es la gamificación, que implica la incorporación de elementos y dinámicas propias de juegos en contextos educativos con el objetivo de aumentar la motivación, la participación y la efectividad del aprendizaje (Dicheva, Dichev, Agre & Angelova, 2015). La gamificación se sustenta en principios psicológicos y pedagógicos ampliamente estudiados, como la teoría de la autodeterminación de Ryan y Deci (2000), que enfatiza la importancia de promover la motivación intrínseca, el compromiso activo y el desarrollo de aprendizajes profundos y significativos.

En este marco, la Universidad Nacional de Rafaela (UNRaf), ubicada en una ciudad reconocida por su fuerte perfil industrial e innovador en la provincia de Santa Fe, Argentina, enfrenta desafíos específicos relacionados con la retención estudiantil y el desarrollo de competencias computacionales avanzadas en sus carreras STEM. Las carreras de Ingeniería en Computación, Licenciatura en Bioinformática y Licenciatura en Agroinformática han registrado durante los últimos años una tendencia decreciente en la matrícula de ingresantes, con un total acumulado de 124 nuevos estudiantes entre 2023 y 2025, junto con indicadores de abandono que motivan la búsqueda de estrategias pedagógicas innovadoras.

El presente trabajo presenta una propuesta metodológica para integrar explícitamente el desarrollo del pensamiento computacional en carreras STEM de la UNRaf mediante estrategias de gamificación. La contribución principal consiste en el diseño de una

secuencia didáctica que articula contenidos disciplinares con actividades orientadas a trabajar de manera explícita las subcompetencias del PC, un esquema de evaluación basado en rúbricas alineadas con dichas subcompetencias, y la presentación de datos contextuales que fundamentan la necesidad de la intervención. El artículo se organiza de la siguiente manera: la Sección 2 presenta el marco teórico; la Sección 3 describe el contexto institucional; la Sección 4 detalla la propuesta metodológica; la Sección 5 presenta los antecedentes empíricos; la Sección 6 discute los resultados esperados y las implicancias; y la Sección 7 ofrece las conclusiones y líneas de trabajo futuro.

2 Marco teórico

2.1 Pensamiento computacional: definición y componentes

El pensamiento computacional ha sido propuesto como un conjunto de habilidades y actitudes que permiten formular problemas y diseñar soluciones de manera que puedan ser ejecutadas por un agente de procesamiento de información, humano o automático (Wing, 2006). Wing lo presenta como una forma de pensar general que puede aplicarse más allá de la programación y que debería ocupar un lugar análogo al de la lectura, la escritura y el cálculo en los sistemas educativos (Wing, 2006; Wing, 2011). Entre sus componentes fundamentales se incluyen la descomposición de problemas complejos en subproblemas manejables, la abstracción de los detalles irrelevantes, el reconocimiento de patrones, el diseño y evaluación de algoritmos y la construcción de representaciones de datos adecuadas (Shute et al., 2017).

Desde una perspectiva educativa, Shute, Sun y Asbell-Clarke (2017) proponen un modelo de pensamiento computacional que organiza habilidades y disposiciones en dimensiones observables, lo que permite diseñar actividades e instrumentos de evaluación alineados con dichas dimensiones. Bocconi et al. (2016) discuten las implicancias de la incorporación del PC en la educación obligatoria, destacando la necesidad de integrarlo en asignaturas disciplinares y no tratarlo solo como un contenido aislado. Denning (2009) aporta una perspectiva crítica al distinguir entre el pensamiento computacional como habilidad general y su aplicación específica en contextos profesionales de computación, enfatizando que la profundidad del desarrollo del PC debe adaptarse al nivel educativo y al campo disciplinar.

En el ámbito de la ingeniería, Zhang, Herman y Wang (2021) discuten el potencial del pensamiento computacional para abordar problemas de alta complejidad, subrayando su relevancia como competencia transversal para disciplinas que requieren modelización, simulación y análisis de sistemas. Esta perspectiva resulta particularmente pertinente para las carreras STEM que combinan fundamentos informáticos con aplicaciones en biología, agronomía e ingeniería, como las ofrecidas por la UNRaf.

2.2 Gamificación en educación: fundamentos teóricos

La gamificación se define como la aplicación de elementos, mecánicas y dinámicas propias de los juegos en contextos no lúdicos, con el propósito de incrementar la motivación, el compromiso y la participación de los individuos (Dicheva et al., 2015). En el ámbito educativo, esta estrategia ha ganado creciente atención como medio para

transformar las experiencias de aprendizaje, pasando de enfoques pasivos y centrados en el docente a modelos activos y centrados en el estudiante.

El sustento teórico principal de la gamificación educativa proviene de la teoría de la autodeterminación (SDT) de Ryan y Deci (2000), que identifica tres necesidades psicológicas básicas cuya satisfacción promueve la motivación intrínseca: autonomía, competencia y relación. Las estrategias gamificadas bien diseñadas abordan estas necesidades al ofrecer opciones significativas (autonomía), desafíos progresivos con retroalimentación inmediata (competencia) y oportunidades de colaboración e interacción social (relación). De acuerdo con Dicheva et al. (2015), los elementos de gamificación más frecuentemente utilizados en contextos educativos incluyen puntos, insignias, tablas de clasificación, niveles, desafíos y narrativas, cada uno de los cuales apunta a diferentes dimensiones motivacionales.

Es importante distinguir entre gamificación y aprendizaje basado en juegos (game-based learning). Mientras que la primera incorpora elementos de juego en actividades educativas existentes, el segundo utiliza juegos completos como vehículos de aprendizaje. Ambas estrategias pueden complementarse y, en la práctica, muchas intervenciones educativas combinan elementos de ambos enfoques (Costa, Martins & Domingos, 2023). En el contexto del presente trabajo, se adopta una perspectiva amplia que integra tanto elementos de gamificación como actividades de aprendizaje basadas en juegos, tanto digitales como analógicos.

2.3 Gamificación y pensamiento computacional: evidencia empírica

Diversas investigaciones respaldan la aplicación exitosa de la gamificación en el desarrollo del PC. Kazimoglu, Kiernan, Bacon y MacKinnon (2012) diseñaron el juego “Program Your Robot”, que requiere aplicar abstracción, pensamiento algorítmico y reconocimiento de patrones. Mediante un diseño cuasiexperimental con estudiantes de secundaria, los resultados mostraron mejoras significativas en habilidades algorítmicas, resolución de problemas y conceptualización abstracta en el grupo experimental, junto con mayor motivación y participación activa.

Kalelioğlu y Gülbahar (2014) exploraron la influencia de técnicas instruccionales basadas en juegos digitales en el desarrollo de habilidades de PC y pensamiento crítico, encontrando mejoras estadísticamente significativas en habilidades analíticas, resolución algorítmica y evaluación crítica. En una línea similar, Kalelioğlu y Gülbahar (2020) documentaron los efectos positivos de la enseñanza de programación mediante Scratch sobre las habilidades de resolución de problemas. Papadakis, Kalogiannakis y Zaranis (2017) analizaron la efectividad de actividades gamificadas en tabletas para desarrollar habilidades numéricas tempranas y PC en niños de preescolar, encontrando mejoras significativas tras una intervención de 14 semanas.

En el nivel de educación superior, Costa, Martins y Domingos (2023) reportaron que un enfoque de aprendizaje basado en proyectos (proyecto CarRace) promovió mejoras en habilidades de PC tales como abstracción, descomposición, reconocimiento de patrones, diseño de algoritmos y depuración en estudiantes de ingeniería en computación. Gomes, Rodrigues y Franco (2020) describieron una intervención basada en el juego TIS-100 que mejoró los modelos mentales de sistemas computacionales y las habilidades de PC en estudiantes de educación media técnica. Moreno-León, Robles

y Román-González (2015) utilizaron la herramienta Dr. Scratch para evaluar y fomentar el PC, encontrando mejoras particularmente en estudiantes con puntuaciones iniciales medias.

La revisión sistemática de Moreno Palma, Hinojo Lucena, Romero Rodríguez y Ramos Navas-Parejo (2024) proporciona evidencia robusta sobre la efectividad de la gamificación en el desarrollo del PC mediante resolución de problemas, reportando mejoras significativas en múltiples estudios y contextos educativos. Robles Carmona, Vergara Oliveros, Giraldo Cardozo y Madera Doval (2024) detallaron una estrategia de enseñanza gamificada STEM+ en la que el 85% de los estudiantes de quinto grado mejoraron sus habilidades de PC y el 90% avanzó en resolución de problemas. Complementariamente, Bers (2018) y Brackmann et al. (2017) han demostrado que las actividades unplugged y digitales gamificadas contribuyen al desarrollo cognitivo y emocional desde etapas educativas tempranas.

2.4 Pensamiento computacional en educación superior e ingeniería

En las carreras de computación e ingeniería, los cursos de Algoritmos y Estructuras de Datos constituyen un eje estructurante del plan de estudios. La literatura en educación en computación ha documentado que estos cursos suelen resultar desafiantes para estudiantes de primer año, entre otros motivos por la necesidad de coordinar múltiples niveles de representación: desde la comprensión informal del problema, pasando por el modelado de datos y operaciones, hasta la implementación en un lenguaje de programación concreto (Shute et al., 2017; Zhang et al., 2021; Grover & Pea, 2013).

Se han identificado dificultades recurrentes tales como la tendencia a abordar los problemas directamente desde el código sin una fase explícita de análisis y diseño, la elección inadecuada de estructuras de datos por falta de criterios comparativos y la poca reflexión sobre la eficiencia y escalabilidad de las soluciones. Estos desafíos se relacionan estrechamente con componentes del PC, en particular con la descomposición, la abstracción y el diseño de algoritmos (Wing, 2006; Shute et al., 2017). En consecuencia, numerosos autores argumentan que los cursos de algoritmos y estructuras de datos constituyen un contexto privilegiado para trabajar de manera explícita dichas competencias (Shute et al., 2017; Bocconi et al., 2016; Zhang et al., 2021).

Grover y Pea (2013), en su revisión del campo en educación obligatoria, destacan la importancia de diseñar secuencias de actividades que aborden explícitamente subhabilidades como la descomposición, la abstracción y el razonamiento algorítmico, así como la necesidad de andamiajes adecuados para estudiantes novatos. Estas contribuciones coinciden en la conveniencia de integrar el PC dentro de asignaturas disciplinares, en lugar de tratarlo como un módulo aislado, lo cual constituye un principio rector de la propuesta presentada en este trabajo.

3 Contexto institucional

La Universidad Nacional de Rafaela es una institución pública relativamente joven, creada en 2014 e iniciando sus actividades académicas en 2015, con una fuerte orientación hacia las tecnologías digitales, la vinculación con el entorno productivo regional y el desarrollo de propuestas formativas innovadoras. Se ubica en la ciudad de

Rafaela, provincia de Santa Fe, Argentina, un polo productivo e industrial reconocido a nivel nacional e internacional por su ecosistema de innovación.

En este marco, la UNRaf ofrece tres carreras STEM con componentes significativos de computación: la Ingeniería en Computación (creada en 2018, con inicio de dictado en 2019), la Licenciatura en Bioinformática y la Licenciatura en Agroinformática (ambas creadas en 2020, con inicio en 2021). Estas carreras comparten asignaturas de programación, algoritmos y estructuras de datos en sus primeros años, lo que constituye un terreno común para la implementación de estrategias transversales de desarrollo del pensamiento computacional.

Los datos de matrícula relevados en los últimos tres años (Tabla 1) muestran una tendencia que amerita atención. Si bien la cantidad total de alumnos se ha mantenido relativamente estable, se observa una disminución progresiva en los ingresantes a Ingeniería en Computación (de 76 en 2023 a 52 en 2025), parcialmente compensada por un crecimiento sostenido en Agroinformática (de 8 a 17 en el mismo período). La Tabla 2 presenta los datos totales de alumnos activos y egresados.

Tabla 1. Ingresantes por carrera STEM en la UNRaf (2023-2025).

Carrera	2023	2024	2025
Ingeniería en Computación	76	65	52
Lic. en Bioinformática	29	27	28
Lic. en Agroinformática	8	12	17
Total ingresantes	113	104	97

Tabla 2. Total de alumnos activos y egresados (2023-2025).

Indicador	2023	2024	2025
Alumnos activos	166	161	155
Egresados	0	0	1

Estos datos, junto con la experiencia docente en las cátedras de algoritmos, estructuras de datos y programación, evidencian la necesidad de implementar estrategias pedagógicas innovadoras que no solo mejoren las competencias técnicas de los estudiantes, sino que también incrementen su motivación y compromiso con las carreras STEM. El proyecto de investigación “Mejorando el Pensamiento Computacional en Ambientes Educativos STEM mediante Proyectos de Gamificación”, reconocido por resolución rectoral de la UNRaf en la convocatoria 2025

del Programa de Incentivos a los Docentes Investigadores, proporciona el marco institucional para el desarrollo de esta propuesta.

4 Propuesta metodológica

4.1 Objetivos de la propuesta

La propuesta metodológica tiene como objetivo general analizar e implementar estrategias de gamificación orientadas a proyectos educativos en ambientes STEM, con el fin de fortalecer significativamente las habilidades del pensamiento computacional. Los objetivos específicos incluyen: (a) identificar y evaluar técnicas y herramientas de gamificación adecuadas para la enseñanza efectiva del PC; (b) diseñar e implementar proyectos gamificados interdisciplinarios que integren elementos tecnológicos, científicos, matemáticos e ingenieriles; (c) evaluar sistemáticamente el impacto de las estrategias en el desarrollo de competencias específicas del PC; (d) generar recursos didácticos innovadores que faciliten la integración curricular del PC; y (e) desarrollar guías y recomendaciones metodológicas basadas en los resultados obtenidos.

4.2 Esquema metodológico general

El proyecto se desarrolla a partir de un esquema metodológico estructurado en cinco etapas claramente definidas, que permiten abordar integralmente el diagnóstico inicial, el diseño, la implementación, la evaluación y la socialización de resultados. La metodología busca asegurar la efectividad, replicabilidad y sostenibilidad de la iniciativa en los contextos educativos específicos de las carreras STEM de la UNRaf y escuelas secundarias de la región.

Etapla 1: Diagnóstico inicial y revisión bibliográfica. Se realiza una revisión bibliográfica extensa para identificar técnicas, herramientas y estrategias gamificadas efectivas en contextos STEM. Paralelamente, se desarrolla un diagnóstico de los niveles iniciales de pensamiento computacional de los estudiantes seleccionados, utilizando instrumentos validados internacionalmente. Esta etapa implica la colaboración con docentes y especialistas en informática educativa para definir criterios claros sobre la situación inicial del estudiantado.

Etapla 2: Diseño de actividades y proyectos gamificados. En base al diagnóstico inicial, se procede al diseño metodológico de actividades y proyectos interdisciplinarios de gamificación orientados al fortalecimiento del PC. Se trabaja en equipos interdisciplinarios integrados por docentes de diversas áreas. Se utilizan herramientas digitales específicas (Scratch, Blockly, MIT App Inventor, Code.org) así como recursos educativos analógicos como juegos de mesa educativos adaptados al contexto regional. Se elaboran guías didácticas detalladas para facilitar la implementación efectiva.

Etapla 3: Implementación de proyectos gamificados. Se ejecutan las actividades diseñadas en aulas STEM previamente seleccionadas. Los estudiantes trabajan en grupos, resolviendo desafíos y proyectos interdisciplinarios gamificados que promueven la participación activa, el trabajo colaborativo y la creatividad. Se proporciona soporte técnico-pedagógico continuo por parte del equipo de investigación.

Etapa 4: Evaluación del impacto. Se combinan enfoques cuantitativos y cualitativos. Se utilizan instrumentos de evaluación específicos para medir cambios en habilidades del PC (pruebas pre-post), encuestas de motivación, entrevistas y grupos focales con estudiantes y docentes participantes. Los datos se analizan mediante técnicas estadísticas y análisis de contenido cualitativo.

Etapa 5: Sistematización y difusión de resultados. Se generan documentos técnicos, académicos y didácticos que permitan la transferencia del conocimiento generado. Los resultados se socializan mediante seminarios, talleres de capacitación docente y eventos académicos.

4.3 Secuencia didáctica para el desarrollo del pensamiento computacional

La secuencia didáctica se organiza en cuatro unidades temáticas que articulan los contenidos tradicionales de las asignaturas con objetivos explícitos de pensamiento computacional. Cada unidad combina clases teóricas breves con actividades prácticas en laboratorio, experiencias gamificadas y espacios de reflexión sobre el proceso de resolución de problemas.

Unidad 1: Introducción al razonamiento algorítmico y descomposición de problemas. Se trabaja con problemas simples que permiten identificar entradas, salidas, restricciones y subproblemas, utilizando pseudocódigo y diagramas de flujo. Se incorporan actividades gamificadas tipo unplugged (juegos de mesa, desafíos de secuenciación) y digitales (actividades iniciales en Scratch y Code.org). El foco está en la descomposición y en la construcción de soluciones paso a paso.

Unidad 2: Estructuras lineales y reconocimiento de patrones. Se introducen arreglos, listas, pilas y colas, enfatizando patrones recurrentes de procesamiento (recorrido, búsqueda, filtrado, acumulación). Se utilizan desafíos gamificados con sistemas de puntos y niveles progresivos, donde los estudiantes deben reconocer similitudes entre problemas y reutilizar esquemas de solución. Se incorporan plataformas como Blockly y actividades con elementos competitivos y colaborativos.

Unidad 3: Estructuras jerárquicas y abstracción. Se trabajan árboles y, en forma introductoria, grafos, poniendo el acento en la abstracción de estructuras de datos que modelan relaciones jerárquicas o redes. Se promueve la capacidad de separar la representación lógica de la implementación concreta. Las actividades gamificadas incluyen proyectos interdisciplinarios con MIT App Inventor y desafíos de modelado con narrativas contextualizadas al entorno regional.

Unidad 4: Análisis básico de complejidad y evaluación de algoritmos. Se introducen nociones elementales de complejidad temporal y espacial, comparando algoritmos alternativos para un mismo problema. Se utilizan actividades gamificadas de tipo competitivo (tablas de clasificación, insignias por eficiencia) donde los estudiantes optimizan soluciones y justifican sus decisiones de diseño. Se discuten compromisos entre simplicidad, eficiencia y uso de recursos.

4.4 Tipos de actividades gamificadas orientadas al pensamiento computacional

La propuesta contempla cinco tipos fundamentales de actividades, diseñadas para abordar de forma explícita las subcompetencias del PC:

a) Actividades de descomposición de problemas. Se presentan problemas contextualizados (gestión de pedidos, administración de turnos, procesamiento de registros agronómicos, análisis de secuencias biológicas) y se solicita a los estudiantes identificar entradas, salidas y restricciones; proponer una descomposición en subproblemas o funciones justificando la división elegida; y elaborar esquemas de módulos o diagramas que muestren la relación entre subproblemas. El énfasis está en la calidad de la descomposición, evaluada mediante rúbricas específicas.

b) Actividades de abstracción y modelado de datos. A partir de descripciones narrativas o casos de estudio, los estudiantes definen entidades y relaciones relevantes, seleccionan estructuras de datos adecuadas argumentando su elección, y especifican operaciones necesarias. Estas actividades buscan que los estudiantes distingan entre el modelo conceptual y sus posibles implementaciones.

c) Actividades de reconocimiento de patrones. Se proponen conjuntos de ejercicios que comparten estructuras de solución, solicitando identificar el patrón común subyacente, generalizar el esquema de solución y comparar variantes en términos de complejidad. El objetivo es desarrollar un repertorio de esquemas reutilizables.

d) Actividades de diseño y evaluación de algoritmos. Para problemas seleccionados, los estudiantes formulan soluciones en pseudocódigo, analizan informalmente la complejidad y comparan su solución con alternativas, discutiendo ventajas y desventajas. Estas actividades conectan el diseño algorítmico con la evaluación de eficiencia.

e) Actividades de reflexión metacognitiva. Al final de ciertas prácticas, se incluyen preguntas de reflexión tales como: ¿qué subproblemas identificaste?, ¿qué abstracciones utilizaste y por qué?, ¿reconociste algún patrón similar a otros problemas anteriores?, ¿cómo evaluarías la eficiencia de tu solución? Estas preguntas se utilizan como insumo para discusión grupal y retroalimentación.

4.5 Estrategias docentes y organización

La propuesta combina varias estrategias docentes: andamiaje incremental, partiendo de ejemplos simples resueltos en conjunto que se complejizan gradualmente; trabajo colaborativo en parejas y pequeños grupos; alternancia entre pseudocódigo y código, enfatizando el trabajo previo en pseudocódigo y diagramas; y discusión de soluciones múltiples, comparando alternativas en términos de subcompetencias de PC. La organización temporal contempla, dentro de cada unidad, un ciclo tipo: introducción teórica breve, actividad guiada centrada en una subcompetencia, práctica de laboratorio gamificada, y reflexión y discusión.

La capacitación del personal docente constituye un componente esencial de la propuesta. Se proveen instancias de formación continua para garantizar que los docentes comprendan los fundamentos de la gamificación, dominen las herramientas tecnológicas seleccionadas y puedan adaptar las actividades a las características específicas de sus grupos de estudiantes.

4.6 Esquema de evaluación

El esquema de evaluación se estructura en torno a tres componentes: evaluación formativa, mediante rúbricas para trabajos prácticos que incluyen criterios explícitos relacionados con descomposición, abstracción, elección de estructuras de datos y claridad del diseño algorítmico; evaluación sumativa, con parciales y trabajos integradores que requieren justificar decisiones de diseño y analizar alternativas; y recolección de datos para investigación, incluyendo cuestionarios de percepción sobre habilidades de PC antes y después de la intervención, y análisis de productos de los estudiantes para identificar evidencias de evolución en las subcompetencias.

La Tabla 3 presenta un esquema de rúbrica para la evaluación de subcompetencias del pensamiento computacional en las actividades gamificadas propuestas.

Tabla 3. Rúbrica de evaluación de subcompetencias del pensamiento computacional.

Subcompetencia	Inicial (1-2)	En desarrollo (3-4)	Avanzado (5)
Descomposición	No identifica subproblemas o lo hace de forma incompleta	Identifica subproblemas pero con dependencias poco claras	Descompone sistemáticamente con subproblemas bien definidos
Abstracción	No distingue entre detalles relevantes e irrelevantes	Identifica parcialmente las abstracciones necesarias	Define abstracciones claras separando modelo de implementación
Patrones	No reconoce similitudes entre problemas	Reconoce patrones pero no los generaliza	Identifica, generaliza y reutiliza patrones eficazmente
Diseño algorítmico	No formula soluciones en pseudocódigo o lo hace con errores críticos	Formula soluciones funcionales pero sin análisis de eficiencia	Diseña algoritmos eficientes y justifica decisiones de diseño

5 Antecedentes empíricos

Con el objetivo de fundamentar empíricamente la propuesta, se realizó una búsqueda sistemática en la base de datos Semantic Scholar, recuperando los 50 artículos más relevantes relacionados con la pregunta de investigación. Tras un proceso de cribado que consideró criterios de inclusión relativos a la presencia de elementos de gamificación, abordaje del pensamiento computacional, población estudiantil, carácter empírico del estudio, contexto educativo STEM y medición de resultados de PC, se

seleccionaron 10 estudios para análisis detallado. La Tabla 4 sintetiza las características principales de los estudios incluidos.

Tabla 4. Síntesis de estudios empíricos sobre gamificación y pensamiento computacional.

Estudio	Nivel educativo	Estrategia	Herramientas	Resultados principales
Costa et al. (2023)	Superior	ABP	Python, Pygame	Mejora en abstracción, descomposición, patrones y depuración
Gomes et al. (2020)	Media técnica	ABJ	TIS-100	Mejora en modelos mentales y habilidades de PC
Moreno Palma et al. (2024)	Primaria / Inicial	Resol. problemas	Robótica, Code.org	Mejoras significativas en PC en múltiples estudios
Moreno-León et al. (2015)	Primaria / Media	Herramienta	Scratch, Dr. Scratch	Mejora en PC, especialmente nivel medio inicial
Robles Carmona et al. (2024)	Primaria	ABJ STEM+	Code.org, Minecraft	85% mejoró PC, 90% resolución de problemas
Kazimoglu et al. (2012)	Secundaria	ABJ digital	Program Your Robot	Mejora significativa vs. métodos tradicionales

ABP: Aprendizaje Basado en Proyectos; ABJ: Aprendizaje Basado en Juegos.

Los hallazgos de la revisión permiten identificar varios factores clave para el éxito de las intervenciones gamificadas en el desarrollo del PC. En primer lugar, la relevancia de los problemas con respecto al mundo real emerge como un elemento facilitador del aprendizaje significativo (Costa et al., 2023). En segundo lugar, el andamiaje progresivo de las actividades resulta crucial, particularmente para estudiantes con niveles iniciales intermedios de PC (Moreno-León et al., 2015). En tercer lugar, la diversidad de herramientas y enfoques permite adaptar las estrategias a diferentes estilos de aprendizaje y contextos de recursos (Moreno Palma et al., 2024). Finalmente, la integración curricular de la gamificación, en lugar de su tratamiento como actividad complementaria, contribuye a la efectividad y sostenibilidad de las intervenciones (Gomes et al., 2020).

Es relevante señalar que la mayoría de los estudios revisados se concentran en educación primaria y secundaria, con menor presencia de investigaciones en educación superior. Esta brecha refuerza la pertinencia de la presente propuesta, que se enfoca

precisamente en carreras universitarias STEM y busca generar evidencia empírica sobre la efectividad de la gamificación en ese nivel educativo.

Un análisis transversal de los estudios permite identificar patrones significativos en las estrategias de gamificación utilizadas. El aprendizaje basado en juegos (game-based learning) fue la estrategia más frecuente, presente en seis de los diez estudios, seguido por el aprendizaje basado en proyectos, la computación desenchufada y los enfoques basados en herramientas específicas. Code.org fue la plataforma más frecuentemente mencionada (tres estudios), seguida por Scratch (dos estudios). Entre las herramientas menos frecuentes, pero igualmente relevantes se encuentran lenguajes de programación como Python, bibliotecas de desarrollo de juegos como Pygame, juegos educativos especializados (TIS-100, Angry Birds, Minecraft) y recursos físicos diversos (kits de robótica, rompecabezas, tarjetas, bloques).

En cuanto a las métricas de engagement, solo tres de los diez estudios reportaron indicadores explícitos de participación y compromiso estudiantil. Robles Carmona et al. (2024) registraron un 90% de satisfacción estudiantil; Romero-Solano, Quevedo-Rojas y Figueroa-Corrales (2023) observaron incrementos en la motivación y el engagement; y Pessoa, Araujo, Andrade y Guerrero (2017) reportaron evaluaciones positivas de la interfaz y el nivel de participación. Esta escasez de mediciones sistemáticas de engagement constituye una oportunidad para la presente investigación, que contempla explícitamente la evaluación de la motivación intrínseca y el compromiso estudiantil como variables de interés.

Finalmente, resulta notable que todos los estudios revisados reportaron resultados positivos en alguna dimensión del pensamiento computacional, si bien la calidad de la evidencia y las metodologías variaron considerablemente. Las habilidades de PC más frecuentemente mejoradas fueron el razonamiento algorítmico, la abstracción, la resolución de problemas y la descomposición. Sin embargo, la heterogeneidad en las herramientas de evaluación utilizadas dificulta las comparaciones directas entre estudios, lo cual refuerza la necesidad de instrumentos estandarizados y contextualizados como los propuestos en esta investigación.

6 Discusión

La propuesta metodológica presentada busca responder a dificultades ampliamente documentadas en cursos iniciales de programación y algoritmos en carreras STEM, asociadas a la articulación entre comprensión de problemas, modelado abstracto y diseño de soluciones eficientes (Shute et al., 2017; Zhang et al., 2021; Grover & Pea, 2013). Al integrar explícitamente las subcompetencias del pensamiento computacional en los objetivos, actividades y criterios de evaluación, se intenta pasar de un enfoque centrado casi exclusivamente en la sintaxis del lenguaje y en la resolución de ejercicios aislados a un enfoque que haga visibles los procesos de descomposición, abstracción, reconocimiento de patrones y diseño de algoritmos (Wing, 2006; Shute et al., 2017).

En comparación con enfoques tradicionales, la propuesta enfatiza la importancia de actividades de reflexión metacognitiva y de discusión de soluciones múltiples, aspectos que la literatura identifica como relevantes para favorecer una comprensión profunda y transferible del razonamiento computacional (Shute et al., 2017; Bocconi et al., 2016;

Grover & Pea, 2013). La alineación entre el marco teórico del pensamiento computacional y las prácticas de aula se corresponde con recomendaciones recientes en educación en ingeniería, que subrayan la necesidad de formar competencias para abordar problemas complejos mediante modelización y análisis sistemático (Zhang et al., 2021).

La incorporación de estrategias gamificadas aporta un valor añadido significativo a la propuesta, al abordar simultáneamente dos dimensiones críticas: el desarrollo de competencias cognitivas específicas del PC y el fortalecimiento de la motivación intrínseca de los estudiantes (Ryan & Deci, 2000; Dicheva et al., 2015). La evidencia revisada sugiere que las intervenciones gamificadas bien diseñadas no solo mejoran habilidades técnicas, sino que también transforman la percepción de los estudiantes sobre las disciplinas STEM, haciéndolas más accesibles y atractivas (Costa et al., 2023; Robles Carmona et al., 2024). Esta dimensión motivacional resulta particularmente relevante en el contexto de la UNRaf, donde los datos de matrícula muestran una tendencia decreciente en los ingresantes que requiere atención.

La articulación con el sistema educativo secundario constituye otro elemento distintivo. La literatura sobre transiciones educativas señala que muchas de las dificultades que enfrentan los estudiantes de primer año en carreras STEM tienen raíces en la formación previa, particularmente en la falta de exposición al pensamiento algorítmico y a prácticas de resolución de problemas estructurados (Grover & Pea, 2013). Al extender la propuesta a escuelas secundarias técnicas de la región, se busca generar un continuo formativo que facilite la transición hacia la educación superior y reduzca las brechas de conocimiento previo que afectan el rendimiento académico inicial. Los docentes de estas instituciones serán capacitados en las estrategias de gamificación y pensamiento computacional desarrolladas en el proyecto, lo cual genera un beneficio directo en términos de calidad educativa regional y fortalece los vínculos interinstitucionales.

Respecto a las herramientas tecnológicas seleccionadas, la combinación de plataformas digitales (Scratch, Blockly, MIT App Inventor, Code.org) con recursos analógicos (juegos de mesa educativos, actividades unplugged) responde a la evidencia empírica que sugiere que la diversidad de modalidades favorece el aprendizaje en estudiantes con diferentes estilos cognitivos y niveles de familiaridad tecnológica (Moreno Palma et al., 2024; Guarda & Goulart, 2018). Las actividades unplugged, en particular, permiten trabajar los fundamentos del pensamiento computacional sin las barreras que pueden imponer las interfaces de programación, facilitando la participación de estudiantes que aún no se sienten cómodos con entornos de código. Esta estrategia de inclusión es especialmente relevante en el contexto de la UNRaf, donde los estudiantes llegan con niveles heterogéneos de conocimiento previo en programación.

No obstante, se reconocen limitaciones del presente trabajo. La implementación efectiva de la propuesta requerirá ajustes progresivos, tanto en la carga de trabajo como en la formación docente, para evitar sobrecargar a estudiantes de primer año. Asimismo, la validación empírica sistemática se plantea como trabajo futuro, dado que el proyecto se encuentra actualmente en sus fases iniciales de diagnóstico y diseño. La capacidad de generalizar los resultados estará condicionada por el tamaño de la muestra

y las características específicas del contexto institucional. Adicionalmente, será necesario desarrollar instrumentos de evaluación que capturen adecuadamente no solo las mejoras en habilidades de PC, sino también los cambios actitudinales y motivacionales que la gamificación busca promover, contemplando las particularidades culturales y educativas del contexto argentino.

7 Conclusiones y trabajo futuro

En este artículo se ha presentado una propuesta metodológica integral para fortalecer el desarrollo del pensamiento computacional en carreras STEM de la Universidad Nacional de Rafaela mediante estrategias de gamificación. La contribución principal radica en la articulación de tres elementos: un marco teórico que integra pensamiento computacional, gamificación y teoría de la autodeterminación; una secuencia didáctica detallada organizada en cuatro unidades temáticas con actividades gamificadas específicas; y un esquema de evaluación basado en rúbricas alineadas con las subcompetencias del PC.

La revisión de diez estudios empíricos proporciona evidencia consistente sobre la efectividad de la gamificación en el desarrollo del PC en diversos contextos educativos, aunque se identifica una brecha significativa en la investigación a nivel de educación superior que este proyecto busca contribuir a cerrar. Los datos contextuales de matrícula y retención de las carreras STEM de la UNRaf fundamentan la urgencia de implementar estrategias innovadoras que mejoren tanto las competencias técnicas como la motivación estudiantil.

Como trabajo futuro, se prevé la implementación gradual de la propuesta en sucesivas cohortes, junto con la recolección sistemática de datos cuantitativos y cualitativos que permitan evaluar su impacto en el rendimiento académico, la retención en las carreras y la evolución percibida de las habilidades de pensamiento computacional. Se proyecta extender el enfoque a escuelas secundarias técnicas de la región, explorar instrumentos específicos de evaluación de PC validados en la literatura y consolidar redes académicas con otras instituciones nacionales e internacionales que trabajan temáticas afines. Actualmente, se trabaja con tres becarios, uno de cada carrera STEM, con el fin de articular las experiencias logradas y poder comparar resultados entre los diferentes programas académicos.

De este modo, la experiencia aquí presentada constituye un primer paso hacia una integración más amplia y sistemática del pensamiento computacional como eje estructurante de la formación STEM, contribuyendo tanto al fortalecimiento académico institucional como al desarrollo integral del territorio de Rafaela y su región.

8 Referencias

- Bers, M. U. (2018). Coding and computational thinking in early childhood: The impact of ScratchJr in Europe. *European Journal of STEM Education*, 3(3), Article 08.
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, D., Engelhardt, A. y Kampylis, K. (2016). Developing computational thinking in compulsory education: Implications for policy and practice. JRC Technical Report, EUR 28295 EN.

- Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A. y Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, 65–72.
- Costa, M. C., Martins, S. G. y Domingos, A. (2023). AS STEM e o pensamento computacional: resolvendo desafios da vida real no ensino superior. REAMEC - Rede Amazônica de Educação em Ciências e Matemática.
- Denning, P. J. (2009). The profession of IT: The problem of computational thinking. *Communications of the ACM*, 52(8), 28–30.
- Dicheva, D., Dichev, C., Agre, G. y Angelova, G. (2015). Gamification in education: A systematic mapping study. *Educational Technology & Society*, 18(3), 75–88.
- Giannakoulas, A. y Xinogalos, S. (2018). A pilot study on the effectiveness and acceptance of an educational game for teaching programming concepts to primary school students. *Education and Information Technologies*, 23(5), 1985–2007.
- Gomes, O. F. M., Rodrigues, W. y Franco, R. (2020). Gamificação e pensamento computacional: análise de uma experiência no ensino médio/técnico-integrado.
- Grover, S. y Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43.
- Guarda, G. y Goulart, I. (2018). Jogos lúdicos sob a ótica do pensamento computacional: experiências do projeto Logicamente. *Anais do XXIX Simpósio Brasileiro de Informática na Educação (SBIE 2018)*.
- Kalelioğlu, F. y Gülbahar, Y. (2014). The effect of instructional techniques on critical thinking and computational thinking. *Eurasia Journal of Mathematics, Science & Technology Education*, 10(6), 571–582.
- Kalelioğlu, F. y Gülbahar, Y. (2020). The effects of teaching programming via Scratch on problem-solving skills. *Informatics in Education*, 19(2), 231–254.
- Kazimoglu, C., Kiernan, M., Bacon, L. y MacKinnon, L. (2012). Learning programming at the computational thinking level via digital game-play. *Procedia Computer Science*, 9, 522–531.
- Moreno-León, J., Robles, G. y Román-González, M. (2015). Dr. Scratch: análisis automático de proyectos Scratch para evaluar y fomentar el pensamiento computacional.
- Moreno Palma, N., Hinojo Lucena, F. J., Romero Rodríguez, J. M. y Ramos Navas-Parejo, M. (2024). Desarrollo del pensamiento computacional “desenchufado” mediante resolución de problemas: una revisión sistemática y meta-análisis. *Revista Investigación Educativa*.
- Papadakis, S., Kalogiannakis, M. y Zaranis, N. (2017). The effectiveness of computer and tablet assisted intervention in early childhood students’ understanding of numbers. *Education and Information Technologies*, 22(5), 1847–1871.

- Pessoa, F. I. R., Araujo, A. L., Andrade, W. L. y Guerrero, D. (2017). T-Mind: um aplicativo gamificado para estímulo ao desenvolvimento de habilidades do pensamento computacional.
- Robles Carmona, E. J., Vergara Oliveros, J. J., Giraldo Cardozo, J. C. y Madera Doval, D. P. (2024). Enfoque STEM+ y gamificación para desarrollar el pensamiento computacional en educación básica. *Acta Scientiæ Informaticæ*.
- Romero-Solano, F. E., Quevedo-Rojas, X. C. y Figueroa-Corrales, E. (2023). La gamificación como estrategia para desarrollar el pensamiento lógico en la resolución de problemas matemáticos. *MQRInvestigar*.
- Ryan, R. M. y Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation. *American Psychologist*, 55(1), 68–78.
- Sáez-López, J. M., Román-González, M. y Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers & Education*, 97, 129–141.
- Shute, V. J., Sun, C. y Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158.
- Souza, O. (2019). JogLog - Jogos de raciocínio lógico para alunos do ensino fundamental: um estudo de caso utilizando gamificação e pensamento computacional. *Anais do XXX Simpósio Brasileiro de Informática na Educação (SBIE 2019)*.
- Tsarava, K., Moeller, K., Pinkwart, N., Butz, M., Trautwein, U. y Ninaus, M. (2018). Training computational thinking through board games: The case of “Robot Turtles”. *International Journal of Serious Games*, 5(2), 25–44.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
- Wing, J. M. (2011). Research notebook: Computational thinking—What and why? Carnegie Mellon University.
- Zhang, L., Herman, J. L. y Wang, F. Y. (2021). Computational thinking in engineering education for addressing complex problems. *International Journal of Engineering Education*, 37(5), 1234–1248.