Revista EIA

UNIVERSIDAD EIA

Aplicando Pensamiento Computacional y Programación en Vivo para crear Juegos Serios en Cursos de Física

Alberto Pacheco-González

1. TecNM Tecnológico Nacional de México

CC BY NC ND

Revista EIA
ISSN 1794-1237
e-ISSN 2463-0950
Año XIX/ Volumen 22/ Edición N.44
Julio - diciembre 2025
Reia4405 pp. 1-27

Publicación científica semestral
Universidad EIA, Envigado, Colombia

Para citar este artículo / To reference this article /
Pacheco-González, A.
Aplicando Pensamiento Computacional y Programación en Vivo para crear Juegos Serios en Cursos de Física.

Revista EIA, 22(44), Reia4405 pp. 1-27
https://doi.org/10.24050/reia.v22i43.1826

Autor de correspondencia:
Pacheco-González, A.
Magíster
Correo electrónico:
alberto.pg@chihuahua.tecnm.mx

Recibido: 19-09-2024
Aceptado:10-06-2025
Disponible online: 01-07-2025

Resumen

Se reporta una propuesta didáctica basada en la integración de estrategias de aprendizaje activo, lúdico, inductivo, supervisado y con retos desvanecidos para asistir en el desarrollo de habilidades del pensamiento computacional orientadas a la resolución de problemas. Esta integración metodológica está organizada en etapas de dificultad creciente escalando los niveles de la taxonomía de Bloom. Se reporta además la aplicación de dicho método para desarrollar un juego serio para modelar fenómenos físicos estableciendo para ello una secuencia de retos desvanecidos para ser resueltos de forma autónoma por los aprendices mediante programación en vivo bajo la supervisión de un mentor. La propuesta se implementó en un curso de Física con estudiantes de cuarto semestre de ingeniería industrial y con estudiantes avanzados de sistemas computacionales, quienes desarrollaron un juego serio para aplicar los conceptos de cinemática. Al evaluar los proyectos se encontró un desempeño equiparable entre ambos grupos y los resultados obtenidos via encuestas indicaron un alto grado de aceptación del método, cambios de actitud favorables hacia la Física, probando así, de forma parcial, la efectividad del método del cirujano enfocado en la forma de resolver problemas de forma supervisada aplicando habilidades del pensamiento computacional mediante la programación en vivo con retos desvanecidos.

Palabras clave: Juegos Serios; Aprendizaje Activo; Aprendizaje basado en problemas; Aprendizaje basado en proyectos; Método Mentor-Aprendiz; Retos desvanecidos; Ludificación; Taxonomía de Bloom; Metacognición; Carga Cognitiva; Programación en Vivo; Enseñanza de la Física; Motores de Juego.

Applying Computational Thinking and Live Coding to Create Serious Games for an Introductory Physics Course

Abstract

A didactic proposal based on the integration of gamified active learning strategies, inductive learning, supervised learning, and faded worked challenges is reported to assist in the development of problem-solving computational thinking skills. This methodological integration is organized in stages of increasing difficulty climbing Bloom's taxonomy levels. The application of this method to develop a serious game for modeling physical phenomena is also reported, establishing a sequence of faded worked challenges solved autonomously by learners through live programming under the supervision of a mentor. The proposal was implemented in a Physics course with fourth-semester industrial engineering students and advanced computer systems students, who developed a serious game to apply kinematics concepts. When evaluating the projects, comparable performance was found between both groups, and the qualitative results obtained via surveys indicated a high degree of acceptance of the method, favorable attitude changes towards Physics, thus partially proving the effectiveness of the surgeon method focused on supervised problem-solving by applying computational thinking skills through live programming with faded challenges.

Keywords: Serious Games; Active Learning; Problem-based Learning; Project-based Learning; Cognitive Apprenticeship; Fading worked examples; Gamification; Bloom's Taxonomy; Metacognition; Cognitive Load; Live Coding; Physics Education; Game Engine.

1. Introducción

La enseñanza tradicional de la Física, basada en la memorización y la transmisión pasiva de conocimientos, no resulta ser adecuada ni efectiva para captar el interés, fomentar el pensamiento crítico, la creatividad y consolidar el aprendizaje significativo de los estudiantes actuales (Marusica & Slisko, 2012). En cambio, el desarrollo del interés por las Ciencias desde edad temprana, así como la incorporación de estrategias de aprendizaje activo y lúdico, pueden contribuir a motivar y mejorar su actitud hacia

Revista EIA https://doi.org/10.24050/reia.v22i44.1826

los conceptos teóricos de la Física, al momento de aplicarlos a la resolución de problemas prácticos, incorporando para ello diversas estrategias didácticas, como lo son: las actividades de campo, los experimentos, las simulaciones, los retos y los juegos (Aondofa & Atindi, 2022; Aprilia, 2023).

Es precisamente en esta dirección que se orienta la presente propuesta metodológica, descrita de forma sintética, aportando la incorporación del desarrollo de videojuegos dentro de la enseñanza de la Física, explorando un enfoque muy dinámico, retador, vivencial, centrado en estrategias de aprendizaje activo, la programación y el pensamiento computacional, buscando desde luego, aplicar conceptos de la Física para el desarrollo mismo de los videojuegos, siempre bajo la supervisión de un mentor, cuidando que de forma gradual, cada estudiante se responsabilice de su propio aprendizaje de forma autoregulada, colaborativa, creativa y lúdica, como se detalla a continuación.

2. Precedentes Teóricos y Didácticos

El **pensamiento computacional** formulado inicialmente por (Wing, 2006), carece de una definición definitiva. Sin embargo, varios autores coinciden en que el pensamiento computacional abarca un conjunto de habilidades y estrategias que sirven para representar y *resolver problemas complejos expresados de forma computacional*, proveyendo una herramienta cognitiva interdisciplinaria clave dentro de la sociedad moderna actual, computacionalmente embebida (Hamidi et al, 2023; Espinal, 2024; Ngadengon, 2024).

La **alfabetización matemática** se refiere a la capacidad de un individuo para formular, aplicar e interpretar modelos matemáticos para resolver problemas de la vida real abarcando competencias como la representación, comunicación matemática, producción de estrategias, razonamiento y argumentación (Altun & Morkoyunlu, 2023). Históricamente, la **matematización** ha sido una base para resolver problemas matemáticos. Para reducir la brecha entre el dominio matemático y el modelado de fenómenos físicos (Azzouzi et al, 2022; Bilbao et al, 2024; Ivanjek et al, 2024) han sugerido la incorporación de herramientas digitales para soportar la

matematización en combinación con el pensamiento matemático, el pensamiento algebraico, el pensamiento computacional y las habilidades STEM.

Para entender mejor la metodología propuesta, resulta indispensable presentar primero su marco conceptual, compuesto por las siguientes definiciones (C_1-$_8$):

Los juegos serios (C_1) son aplicaciones computacionales que además de ofrecer esparcimiento y diversión integran intenciones pedagógicas, informativas y de interacción en un videojuego para transformarlo en una herramienta orientada al entrenamiento y el aprendizaje (Ben Rebah & Ben Slama, 2019). Los juegos serios permiten fortalecer la motivación, creatividad, resolución de problemas, colaboración y participación de los estudiantes (Ben Rebah & Ben Slama, 2019; Tomalá et al, 2020; Aondofa & Atindi, 2022; Gurbuz & Celik, 2022; Van der Linder et al, 2024). En el diseño de un juego serio se recomienda alinear la dinámica del juego con los objetivos de aprendizaje, sin olvidar además la incorporación de un alto nivel de interacción y dinamismo (Gurbuz & Celik, 2022; Van der Linder et al, 2024).

La teoría de la carga cognitiva (C_2) estudia la capacidad cognitiva humana integrando un conjunto de principios instruccionales y guías basadas en evidencias que son de utilidad para analizar, diseñar métodos, procesos y ambientes de aprendizaje capaces de explotar las limitaciones cognitivas dosificando los distintos tipos de carga de trabajo: a) la carga intrínseca (*intrinsic cognitive load*) viene dada por la complejidad misma de la tarea, i.e. cantidad de elementos interactivos que deben coordinarse al mismo tiempo en la memoria de trabajo; b) la carga pertinente o relevante (*germane cognitive load*) considera ciertos elementos que serán de ayuda para construir modelos mentales más complejos y robustos para lograr un mejor desempeño probando distintos contextos; c) la carga ajena o irrelevante (*extraneous cognitive load*) son aquellos elementos que distraen, retardan u obstaculizan el aprendizaje, e.g. redundancias, ambigüedades, errores u omisiones en materiales educativos mal diseñados. De acuerdo con esta teoría, un buen diseño instruccional debe dosificar, secuenciar y modularizar la carga intrínseca de una actividad de aprendizaje, buscando reducir los

Revista EIA https://doi.org/10.24050/reia.v22i44.1826

elementos distractores y las cargas cognitivas irrelevantes, liberando así más memoria de trabajo para proporcionar elementos que promuevan el desarrollo de nuevas habilidades y esquemas mentales (carga relevante). Además, (Clark, Nguyen & Sweller, 2006) sugieren un diseño de actividades dinámicas, flexibles, fluidas, autoreguladas, retadoras y motivadoras acordes al *expertise* del aprendiz.

La metacognición (C_3) se referiere a la conciencia que el individuo tiene de su propio proceso de pensamiento, involucrando: 1) el conocimiento y autoevaluación del proceso cognitivo; 2) la autoregulación del proceso cognitivo y conductual para lograr un objetivo (Vorhölter, 2021). En (Wilson & Clarke, 2003) se reporta estudio con estudiantes con habilidades expertas usando la metacognición para monitorear su proceso continuo y reflexivo de resolución de problemas, siendo además capaces de explicar sus estrategias en voz alta.

Los ejemplos resueltos (C_4) son demostraciones o guías paso-a-paso que ilustran cómo efectuar una tarea o resolver un problema, con el objetivo de incrementar el *expertise* del aprendiz de forma más acelerada que un ejercicio de práctica (Spieler et al, 2020). Existen distintos tipos de ejemplos resueltos: 1) el ejemplo resuelto guiado (*worked example*), más orientado al aprendiz novicio; 2) par ejemplo-problema (*worked example-problem pair*) son secuencias de ejemplos resueltos emparejados con ejercicios, más orientado al aprendiz intermedio; 3) ejemplo resuelto desvanecido (*fading worked example*), donde gradualmente se incorporan menos ejemplos resueltos, a la vez que se agregan más pasos como problemas a resolver buscando que el aprendiz más avanzado demuestre su trabajo como evidencia de aprendizaje, incentivando además un procesamiento más profundo, ejercitando y consolidando un esquema mental en vez de repetir los pasos vistos con anterioridad (Chen et al, 2023).

William Halsted estableció el primer programa de residencias para cirujanos en 1890 (Kotsis & Chung, 2013), proveyendo un método efectivo para guiar y formar cirujanos bajo la tutela y supervisión de un experto que expone sus decisiones y retroalimenta al aprendiz en cada etapa formativa hasta que dicho aprendiz sea capaz de lograr realizar de forma autónoma una cirugía completa.

Por otra parte, de acuerdo con (Rowley, 1999) un buen mentor es apasionado, comprometido, empático, didáctico, versátil, optimista y se supera de forma constante (*lifelong learner*). **El método de Halsted** (C_5), tradicionalmente parafraseado como "*see one, do one, teach one*" y más recientemente como "*see many, do many with supervision, teach many with supervision*" implica una creciente adquisición de competencias y responsabilidades hasta culminar en un aprendiz más apto y autónomo. Este método también conocido como "*the match*", resulta cada vez más pertinente para desarrollar un conjunto portable de competencias (*portable skill set*) que pueda ser transferido y aplicado a un mayor números de oportunidades y desafíos dentro del mundo profesional actual (Steelman et al, 2023).

El entrenamiento basado en aprendizaje supervisado, también denominado **mentor-aprendiz** (C_6), se refiere al proceso de transferir conocimientos de una persona (mentor) a otra (aprendiz), mostrando qué hacer y cómo hacerlo. La *teoría del aprendiz cognitivo* introducida por (Collins et al, 1987) implica que el mentor haga visible al aprendiz sus procesos cognitivos. Dicha teoría abarca seis métodos de instrucción: modelado, entrenamiento, andamiaje (*scaffolding*), articulación, reflexión y exploración.

La programación en vivo o ***live coding*** (C_7) implica que un instructor programe en tiempo real frente a una clase. A diferencia de explicar código prescrito y fijo, este método tiene como objetivo modelar estrategias, técnicas y heurísticas para resolver problemas. Cuando se aplica este método, se recomienda balancear la comprensión del programa (método tradicional) con las habilidades para construir programas (Shah et al, 2023). Existen variaciones, la más usual se basa en el modelado (*teoría del aprendiz cognitivo*), donde el mentor verbaliza sus procesos de pensamiento y explica el propósito que hay detrás de cada estrategia elegida conforme resuelve un problema. Por su parte el entrenamiento y *andamiaje* pueden aplicarse cuando se solicita al aprendiz pasar al frente del grupo a resolver una tarea específica bajo supervisión retroalimentada del mentor.

La taxonomía de Bloom (C_8) adaptada para la Educación en la Computación (ACM, 2023) sugiere una colección de verbos para cada nivel, tales como: 1) recuerda: busca, localiza, identifica, memoriza,

Revista EIA https://doi.org/10.24050/reia.v22i44.1826

define, enumera, elige; 2) comprende: anota, explica, discute, clasifica, describe, comenta, ejemplifica; 3) aplica: programa, construye, resuelve, usa, opera, configura, edita, modifica, manipula; 4) analiza: modela, simula, detecta, generaliza, integra, compara, descompone, examina, actualiza; 5) evalúa: asesora, debate, argumenta, defiende, critica, depura, prueba, optimiza, adapta, decide, verifica, estima; 6) crea: diseña, colabora, planifica, construye, programa, genera, inventa, planifica, visualiza.

3. Materiales y Métodos

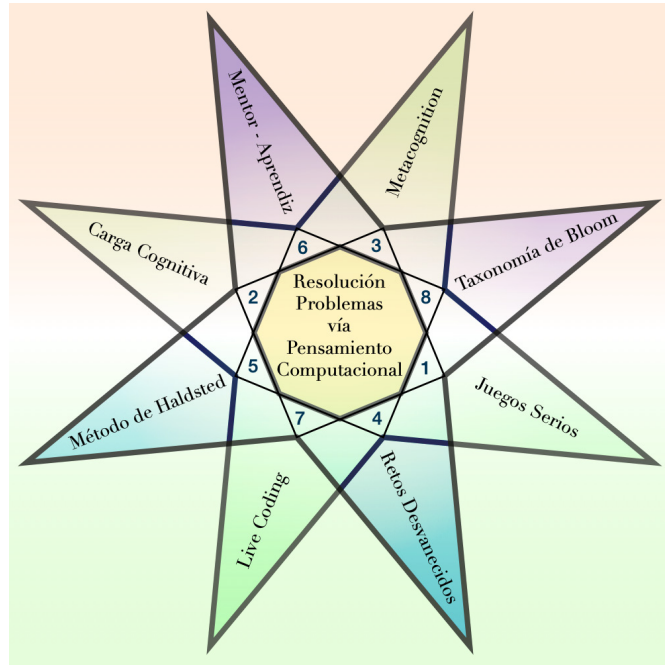
Para efectos metodológicos dentro del presente trabajo, cuya conexión con los conceptos teóricos descritos anteriormente (definiciones $C_{1\text{-}8}$) será indicada posteriormente (utilizando dicha notación según corresponda), explora las siguientes preguntas de investigación:

- RQ_1. ¿es posible aplicar el pensamiento computacional para simular leyes de la Física desarrollando videojuegos con estudiantes que no saben programar?
- RQ_2. ¿la estrategia de aprendizaje activo mejora la actitud, rendimiento y/o motivación de un estudiante en un curso de Física?
- RQ_3. ¿la mentoria y los retos desvanecidos contribuyen a incrementar la autonomía, metacognición y/o habilidades para resolver problemas?
- RQ_4. ¿es posible dentro de una sana competencia balancear a la vez la creatividad, participación activa y colaboración entre los miembros del equipo?

La exploración didáctica cuasi-experimental fue llevada a lo largo de los semestres académicos enero-junio 2024 y enero-junio 2025, dedicando dos horas semanales con: 1) grupos de tanto la materia de videojuegos de la carrera de ingeniería en sistemas computacionales de último semestre, donde dichos estudiantes contaban con experiencia en programación (grupo de control); 2)

grupos del curso de Física de cuarto semestre de ingeniería industrial sin experiencia en programación (grupo experimental). En el grupo de control se impartieron las clases de manera tradicional y en el grupo experimental de Física se aplicó una integración de los conceptos $C_{1\text{-}8}$ mencionados en la sección anterior, que en delante se denominará **método del cirujano con retos devanecidos** (Fig. 1). En ambos grupos cada estudiante de forma individual desarrolló un juego serio (aprendizaje basado en proyectos), usando un motor de juegos multiplataforma de software libre (Godot, 2024), y además, se les permitió consultar todo tipo de recurso: foros (Stack Overflow), video tutoriales (YouTube), repositorios de código (GitHub) y chatbots (ChatGPT, DeepSeek). Para el grupo de control, el instructor impartió el curso de forma tradicional explicando ejemplos preescritos por tema, sin aplicar el método del cirujano y usando una plataforma *b-Leaning* para gestionar el curso, proporcionar ejercicios y materiales.

Figura 1. *Marco conceptual del método del cirujano con retos desvanecidos.*



Por otra parte, dentro del grupo experimental, con la intención de aplicar nociones conceptuales propias de los cursos de Física, se

Revista EIA https://doi.org/10.24050/reia.v22i44.1826

propuso desarrollar como proyecto de clase un juego serio (C_1) para simular dichos principios via la matematización, mismos que fueron problematizados y descompuestos en una serie de retos semanales (C_2, C_4). Al igual que el grupo de control, fuera de clase (*aula invertida*), los aprendices del grupo experimental consultaron todo tipo de recursos para intentar resolver cada reto semanal, que era abordado en una de las sesiones semanales aplicando el método del cirujano (C_5), donde el mentor-cirujano (C_6), luego de plantear el problema o reto (C_4), descomponiéndolo en partes (tareas, y cada tarea en pasos), explicó y resolvió ciertos pasos del problema programando en vivo (C_7) y pensando en voz alta (cirugía), dando además la oportunidad de pasar al frente a un aprendiz para resolver, pensando también en voz alta, una etapa de la cirugía bajo la supervisión del mentor (C_5, C_6), mientras el resto de los aprendices tomaban notas y hacían preguntas sin transcribir código. Durante dicho proceso, no se penalizó ningún error y se incentivó la colaboración, la creatividad y la competencia entre los aprendices, ya que cada cirugía exitosa de un aprendiz otorgaba puntos extra en la calificación (*gamification*). Luego de resolver el problema, por lo general, el aprendiz cambiaba al rol de mentor cuando asesoraba a sus compañeros respecto a la técnica usada durante su cirugía en vivo (C_5, C_7). La metodología se divide en tres etapas que regulan de manera gradual la complejidad de los problemas o retos desvanecidos (C_2, C_3, C_4, C_8), detallados en la Tabla 1, durante el desarrollo del juego serio (ilustrados de forma sintética en la Figura 2):

a) **Etapa de Principiante.** Se introdujo el entorno de programación y el motor de juegos, así como las nociones de programación para el desarrollo de un videojuego por medio de ejercicios resueltos y retos desvanecidos (C_4) abordando problemas simples con baja carga cognitiva intrínseca (C_2), consistentes en una sola tarea con sólo un par de pasos, buscando que el aprendiz repitiera el ejercicio, correspondientes a los primeros niveles de taxonomía de Bloom (C_8). Por ejemplo, luego de explicar el entorno del motor de juegos y el primer avance de la simulación de caída libre de objetos, habiendo encargado que

los alumnos buscaran y seleccionaran sus propios sonidos y gráficos, luego de explicar las propiedades y diferencias entre objetos rígidos y estáticos, se invitó a los alumnos a pasar al frente para modificar la imagen del personaje y ajustar la silueta del nuevo personaje para detectar colisiones. Al concluir la cirugía (C_5), se premió (*gamification*) y calificó el desempeño de cada paso de la cirugía registrándose en un tablero dentro de la plataforma de gestión del curso. Al agotarse el tiempo de sesión, los pasos subsecuentes quedan como ejercicios de tarea y se retoman en la siguiente sesión, incentivando así a que el aprendiz repase, investigue y practique fuera de clase (C_3), con la intención de prepararse para participar en las próximas cirugías.

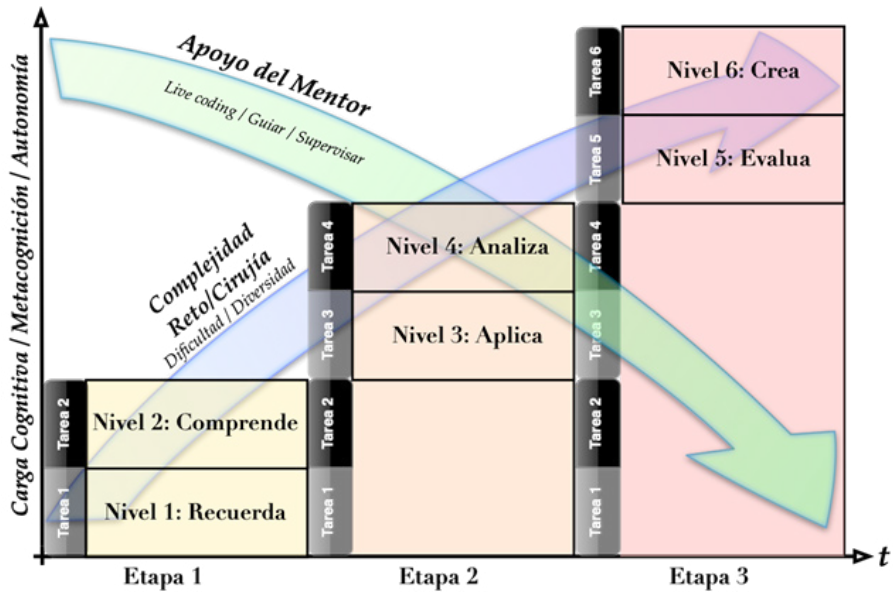
b) Etapa Intermedia. Se incrementó un tanto la dificultad de los retos (dos tareas con dos pasos por reto), incorporando cada vez más carga cognitiva relevante (C_2) sobre los aspectos relacionados con el desarrollo del juego serio (C_1). En esta etapa se consideran principalmente los niveles intermedios de la taxonomía de Bloom (C_8), donde el aprendiz internaliza, reafirma, analiza y aplica los conceptos y técnicas básicas: operando, manipulando y configurando el motor de juegos; editando y modificando código (Bloom, nivel 3); analizando, examinando, comparando, descomponiendo y actualizando el código del proyecto (Bloom, nivel 4). Por otra parte, durante las sesiones de programación en vivo (C_7), se promueve un aprendizaje metacognitivo más intenso (C_3) para resolver problemas y desarrollar algoritmos (Bloom, nivel 3), refinando, generalizando, mejorando (*refactoring*) y depurando (*debugging*) dichos algoritmos del videojuego con la asistencia del mentor y consultando videos, *chatbots*, páginas, e incluso a sus propios compañeros.

c) Etapa Avanzada. Las últimas etapas de desarrollo del juego serio (C_1) abarcaron problemas más complejos (tres tareas con tres pasos por reto), utilizando menos ejemplos resueltos (C_4). Durante las sesiones de programación en vivo (C_7), el mentor se enfocó más en supervisar cirugías y atender dudas de los aprendices, promoviendo además la creatividad, especialización, autoregulación y construcción del propio conocimiento (C_3) en

Revista EIA https://doi.org/10.24050/reia.v22i44.1826

cada aprendiz de una manera dinámica y fluida. Desde el punto de vista de competencias para el pensamiento computacional se presto mayor atención a la integración de técnicas avanzadas, como la refactorización, manejo de versiones, documentación, respaldos y depuración del código para mejorar la eficiencia, estabilidad, jugabilidad y usabilidad del juego serio (C_1) integrando los últimos niveles de la taxonomía de Bloom (C_8).

Figura 2. *Etapas principales del proceso de aprendizaje supervisado con programación en vivo y retos desvanecidos (cirugías).*



Los últimos niveles de la taxonomía de Bloom aplicados en esta etapa avanzada fueron los siguientes:

- Nivel 5 "Evaluar" de la taxonomía de Bloom (C_8):
  i. Adapta, evalua y verifica de forma autoregulada y declarativa.
  ii. Prueba, depura, calibra y optimiza de forma activa y procedimental.
  iii. Asesora, debate, argumenta y defiende de forma colaborativa.

- Nivel 6 “Crear” de la taxonomía de Bloom (C_8):

  i. Diseña, colabora y planifica de forma colaborativa.

  ii. Crea, inventa, construye, genera y programa según la aptitud, preferencia y disposición del aprendiz.

  iii. Planifica, organiza, sintetiza, documenta, reporta proyecto final.

Como se muestra en la Tabla 1, el diseño del juego serio (C_1) empata una secuencia de retos $R_{0\text{-}10}$ (C_4) con varios temas del curso de Física ($T_{1\text{-}12}$) y con el respectivo aprendizaje de diversas herramientas del entorno del motor de juegos, así como diversas técnicas de programación ($H_{1\text{-}12}$) requeridas para crear los componentes del juego serio.

Por ejemplo, en las primeras dos cirugías (C_4, C_5) efectuadas por el mentor (Tabla 1), se ilustran los temas $T_{1\text{-}2}$ (características de objetos estáticos y rígidos) y las herramientas $G_{1\text{-}2}$ (partes del entorno y el manejo de recursos); en las cirugías iniciales (C_4, C_5, C_7) no se requirió la participación de los aprendices (C_6), sin embargo, se enfatizó la importancia de saber tomar notas e identificar los términos clave (C_2), considerando el primer nivel de la taxonomía de Bloom (C_8).

Como parte de la metodología, para validar la actitud y opiniones de los estudiantes ante esta exploración y experiencia didáctica se consideraron los siguientes **instrumentos**:

In_1: Encuesta diagnóstica usando escala de Likert.

In_2: Evaluación del proyecto (planeación, lista cotejo) (Tabla 1 y 2).

In_3: Encuesta a los jugadores invitados a probar juego serio (Fig. 6).

In_4: Encuesta de opinión final (Fig. 7).

Revista EIA https://doi.org/10.24050/reia.v22i44.1826

Tabla 1. *Planeación de los retos por etapa cubriendo un tema y herramienta específica, abordados como cirujías tanto para el mentor como los aprendices hasta concluir el desarrollo del juego serio.*

T/H/C Etapa/Reto	E_1 / R_0	E_1 / R_1	E_1 / R_2	E_2 / R_3	E_2 / R_4	E_2 / R_5	E_2 / R_6	E_3 / R_7	E_3 / R_8	E_3 / R_9	E_3 / R_{10}
Temas / Herramientas No. Cirujías: Mentor/ Aprendiz	T_{1-2} / H_{1-2} 2 / 0	T_{3-4} / H_3 2 / 1	T_5 / H_4 2 / 1	T_{1-6} / H_5 1 / 2	T_6 / H_6 1 / 2	H_7 1 / 2	T_7 / H_8 2 / 2	T_9 / H_9 1 / 3	T_{10} / H_{10} 2 / 10	T_7 / H_{11} 0 / 10	H_{12} 1 / 10

Tema	Concepto de Física aplicado	Herr.	Herramientas de Godot aplicada
T_1	Cuerpos estáticos y rígidos	H_1	Manejo del entorno: menus, editores, browsers
T_2	Caída libre y cte. gravedad	H_2	Manejo de recursos (assets, imágenes)
T_3	Masa, veloc. inicial, elasticidad	H_3	Manejo de propiedades (peso, giro, damping)
T_4	Torque angular, fricción, amortig.	H_4	Scripts, variables, funciones (rnd), eventos
T_5	Matematizar (desp, dim, prop, cte)	H_5	Detección de colisiones y control teclado (cursor)
T_6	Cálculo de tiempo	H_6	Score y layouts
T_7	Simulación: visual, calib, score	H_7	Sonidos, timers y efectos
T_8	Cálculo de distancia	H_8	Manejo de niveles y variables globales
T_9	Mov. circular uniforme. Torque angular	H_9	Controles UI (botón, lista, contenedor)
T_{10}	Preguntas de Repaso	H_{10}	Niv 2: generador preguntas y opc. múltiples al azar
T_{11}^*	Manejo de fuerzas	H_{11}	Niv 3: Reusar nivel 1 con distancia al azar
T_{12}^*	Plano inclinado, tiro parabólico, etc.	H_{12}	Pantalla de inicio. Nivel 4 (pregs). Pantalla final

Figura 3. *Reto de ajustar distancia de impacto en objeto en caída libre.*

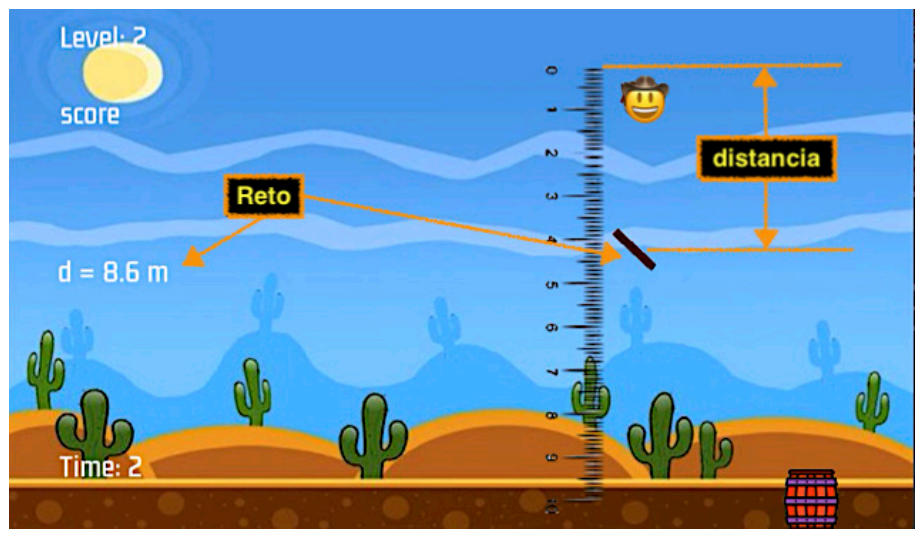
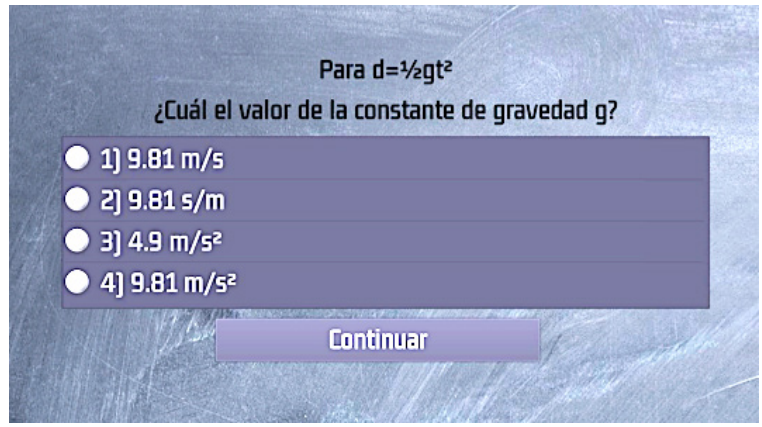


Figura 4. *Preguntas teóricas al azar para saltar al siguiente nivel del juego serio.*



Desde una perspectiva metódica (Figs. 1 y 2), en las primeras cirugías (C_4, C_5), gracias al motor de juegos (C_1), se logró simular la caída libre del objeto sin escribir una línea de código. Lógicamente, los siguientes retos (C_4) consistieron en iniciar la caída del objeto hasta pulsar una tecla, mover el peldaño con las teclas cursor, detectar la colisión del objeto contra el peldaño (tiempo y sonido) y el piso (eliminar objeto y reiniciar). En contraste, en la etapa final (retos $R_{7\text{-}10}$ de la Tabla 1) predominaron cirugías realizadas por aprendices, exigiendo una mayor preparación (aprendizaje autónomo, C_3), una mayor dosis de carga relevante (C_2) y varias habilidades avanzadas del pensamiento computacional (abstraer, diseñar, mejorar, documentar y depurar algoritmos).

El juego serio, descrito a mayor detalle en (Pacheco, 2024), simula la caída libre de objetos despreciando la resistencia del aire y partiendo del reposo (Fig. 3). Por otra parte, la intención didáctica del juego consiste en descubrir la manera de calcular la distancia que recorre un objeto en caída libre en función del tiempo, mediante:

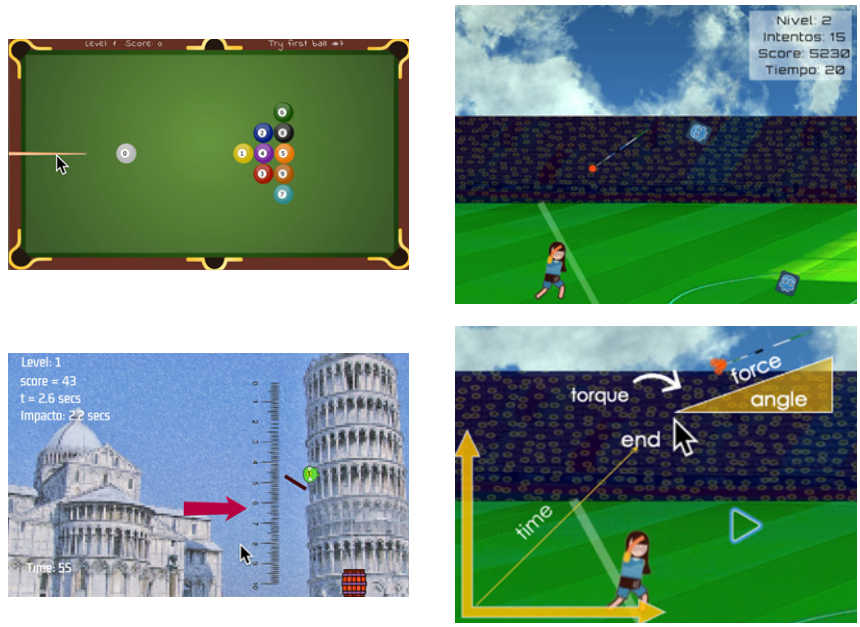
$$d = \frac{g}{2}t^2$$

Para ello, en el primer nivel del juego, se intentó recrear el experimento de caída libre de Galileo (Van Hemmen, 2021),

Revista EIA https://doi.org/10.24050/reia.v22i44.1826

donde primero se deduce la proporcionalidad directa que existe entre la distancia recorrida y el tiempo transcurrido (t_{reto}). Esto se facilita proporcionando tiempos de forma aleatoria (reto), con la intención de que el jugador desplace una repisa con el cursor hasta aproximarse a la distancia que corresponda al tiempo de caída t_{reto}, para luego, premiar según sea la magnitud del error = $|t_{reto} - t_{real}|$ (Fig. 3). Estas jugadas se repiten hasta agotar un minuto de juego, apareciendo luego una pregunta teórica al azar para confirmar el logro del objetivo de aprendizaje (Fig. 4). Se avanza ante una respuesta correcta o se retorna al nivel anterior. En el siguiente nivel se determina el valor de la constante g. De forma análoga, se genera una distancia aleatoria (d_{reto}) y se mueve el peldaño para aproximarlo a la distancia buscada (Fig. 3). Al agotarse el tiempo, una pregunta al azar verifica lo aprendido (Fig. 4). Si la respuesta es correcta se termina el juego reportando la puntuación total alcanzada, en caso contrario, se regresa al nivel anterior. Durante el semestre enero-junio 2025 fue posible desarrollar juegos con nuevas temáticas: lanzamiento de jabalina para ilustrar el tiro parabólico y juego del billar para vectores de fuerzas aplicadas y resultantes, torque angular, inercia y fricción (Figura 5).

Figura 5. *Ejemplo de pantallas de juego serio desarrollado en cursos de Física.*



Ejemplo tiro parabólico: https://youtu.be/2vQakhKPBgc

4. Resultados

En el inicio del curso de Física se aplicó una encuesta diagnóstica (In_1) consistente en 5 preguntas en escala de Likert 1=desacuerdo, 5=de acuerdo, n_1=10 (2024), n_2=30 (2025), de donde destacan los siguientes resultados (promedio n_1 y n_2):

1. La mitad del grupo cursa por segunda vez el curso de Física (50 % repetidores).
2. Lo que más les atrae de un videojuego es: la calidad del videojuego (Likert 4.4); jugar y competir (Likert 4.3); lo llamativo, original y creativo del juego (Likert 4.3).
3. Según los alumnos repetidores, su bajo rendimiento en Física se debió a: dificultad para comprender el uso de fórmulas para resolver problemas (80%), conceptos abstractos o aburridos (30%) y malas experiencias con métodos de enseñanza (30%).
4. La mayoría no ha programado nunca o muy poco (66%).
5. Los estudiantes comentaron: “explicar de forma clara, simple, dinámica, visual, colaborativa y práctica”, “enseñar con ejercicios prácticos donde todos interactúen”, “ayudar a los alumnos a resolver la menor duda”.

Al final se evaluó el proyecto del juego serio con una lista cotejo (In_2, Tabla 2). En ambos grupos todos los alumnos terminaron su juego serio, excepto un estudiante avanzado del grupo de sistemas, cuyo juego presentó errores de forma ocasional debido al uso de un algoritmo complicado e inconcluso. Todos los estudiantes invitaron a niños a que corrieran su juego y reportaran su nivel de aceptación, opiniones y retroalimentación (In_3). Al final los estudiantes participantes en ambos cursos de Física (2024-2025), contestaron el instrumento (In_4) (promedio ambos grupos n=40, Fig. 6):

1. Edades: 13-17 años: 56%, 18-24 años: 39%, 12 o menos: 5%;
2. Sexo: 50% mujeres, 50% hombres.

Revista EIA https://doi.org/10.24050/reia.v22i44.1826

3. Escolaridad: 44% preparatoria, 28% universidad, 22% secundaria, 6% primaria.
4. Ha cursado alguna materia de Física: 72%.
5. El 100% tiene algún equipo propio donde puede jugar videojuegos.
6. Le gustan los videojuegos? Sí: 83.3%, tal vez: 11.1%, no: 5.6%.
7. Le gusta la Física? Sí: 38.9%, no: 38.9%, tal vez: 22.2%.
8. Cuántas veces que jugaste? 5 o menos veces: 83%, más de 5 veces: 17%.
9. Tuvo fallas el juego? Ninguna: 89%, una falla: 11%.
10. Te pareció interesante o divertido el juego? Sí: 72%, neutral: 16.7%, no: 11.1%.
11. Se simula correctamente la caída del objeto? Sí: 83%; neutral: 11%, no: 6%.
12. Entendiste cómo lograr hacer el mayor número de puntos? Sí: 89%.
13. Te pareció difícil o confuso de jugar? No: 66.7%, tal vez: 27.8%, sí: 5.6%.
14. Llegaste a pasar al último nivel? Sí: 94.4%.
15. Fueron difíciles las preguntas del juego? No: 44.4%, tal vez: 33.3%, sí: 22.2%.
16. Fue difícil ganar? No: 44.4%, tal vez: 27.8%, sí: 27.8%.
17. Aprendiste algo de Física con el juego? Sí: 61.1%, tal vez: 27.8%, no: 11.1%.
18. El juego sería útil para la clase de Física? Sí: 83.3%, no: 11.1%, tal vez: 5.6%.
19. Te gustaría programar videojuegos? Sí: 66.7%, no: 22.2%, tal vez: 11.1%.
20. Sabías que este videojuego lo hizo un alumno en su curso de Física? Sí: 67%.
21. Te gustaría saber cómo se hizo el videojuego? Sí: 94.4%.

Figura 6. *Algunos de los resultados obtenidos en la encuesta efectuada a los jugadores (In3).*

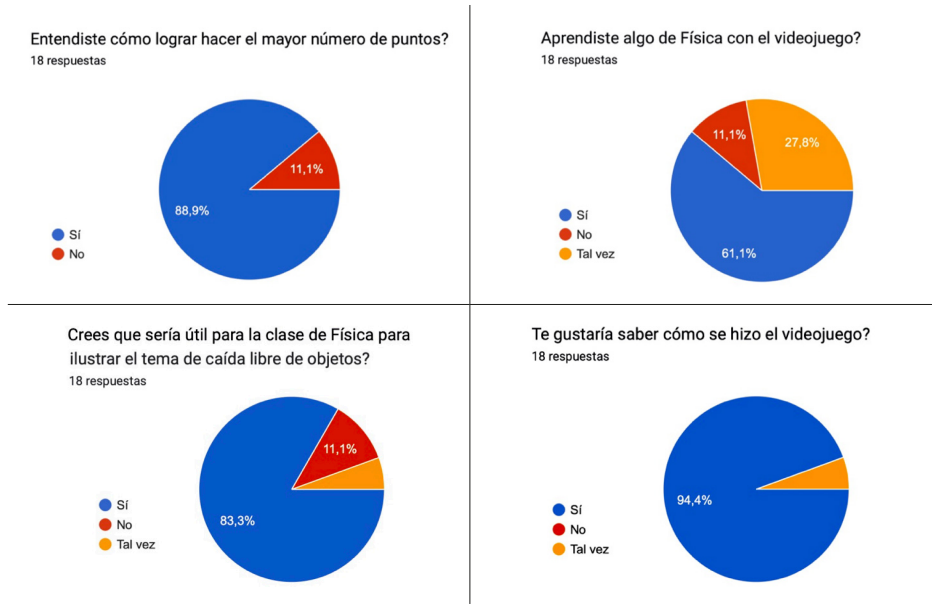
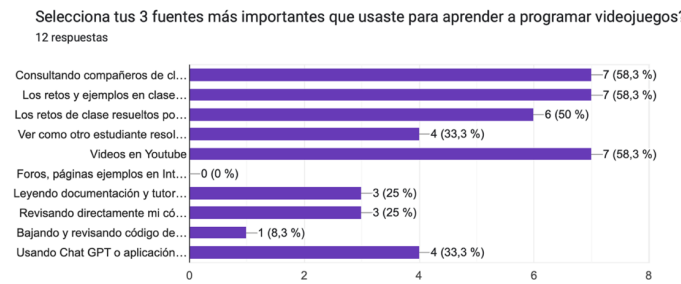
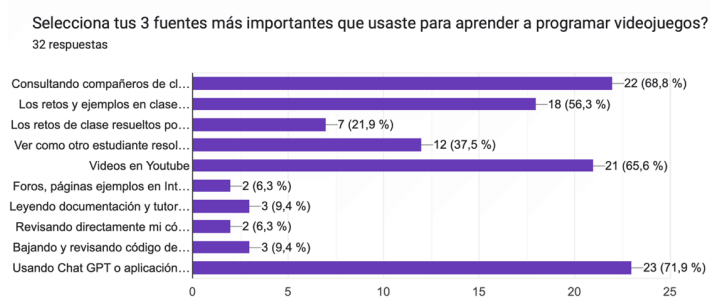


Figura 7. *Fuentes de información más consultadas por estudiantes para ciclos 2024 y 2025.*

a) Ejemplo pregunta encuesta In_4 (n=10) que cursaron Física en mayo de 2024.



b) Ejemplo pregunta encuesta In_4 (n=30) que cursaron Física en mayo de 2025.



Revista EIA https://doi.org/10.24050/reia.v22i44.1826

Respecto a la encuesta final (In_4), promediando ambos grupos (n=40 estudiantes), destacan las siguientes resultados:

1) Antes de iniciar el curso:
   - ¿Sabías programar? No: 50%, muy poco: 42%, regular: 8%.
   - ¿Has programado videojuegos? No: 75%, muy poco: 25%.
2) Motor de juegos y videojuego:
   - Dificultad uso de herramientas (primer mes): Media=2 (1=Muy difícil, 5=Muy fácil)
   - Dificultad uso de herramientas (final): Media=3 (1=Muy difícil, 5=Muy fácil)
   - Dificultad desarrollo juego: Media=2.8 (1=Muy difícil, 5=Muy fácil)
   - ¿Crees que tu videojuego sea educativo? Media=4.33 (1=No, 5=Sí)
3) Experiencias de aprendizaje:
   - ¿Las cirugías fueron de ayuda para aprender? Media=4.2 (1=Nada, 5=Bastante)
   - ¿Qué tanto aprendiste de programación? Media=4.3 (1=Nada, 5=Bastante)
   - Principales fuentes de aprendizaje:
     i. Los retos y ejemplos en clase explicados por el maestro (58%)
     ii. Consultando compañeros de clase (58%)
     iii. Videos en Youtube (58%)
     iv. Los retos de clase (cirugías) resueltos por mi mismo (50%)
     v. Ver como otro estudiante resolvía el problema (33%)
     vi. Usando ChatGPT o aplicación de AI (33%)
   - ¿Mejoraste tus habilidades de programación? Media=4.25 (1=Nada, 5=Bastante)
   - ¿Aprendiste a resolver problemas? Media=3.67 (1=Nada, 5=Bastante)
   - ¿La experiencia de desarrollar un videojuego fue? Media=4.33 (1=Negativa, 5=Positiva)

- ¿La experiencia te permitió expresar tu creatividad? Media=4.17 (1=Nada, 5=Bastante)

4) Opiniones:

- ¿Evaluar la clase con un videojuego fue? Media=4.17 (1=Injusta, 5=Justa)
- ¿Has comentado a otros tu experiencia? Media=4 (1=Nada, 5=Bastante)
- ¿Te gustaría programar en otras materias? Media=4.33 (1=Nada, 5=Bastante)
- ¿Te gustaría aprender a programar antes de la Universidad? Sí: 83%, tal vez=17%, no=0
- ¿Al programar eres productivo y competitivo? Media=4.5 (1=Nada, 5=Bastante)
- Comentarios finales:
- "Nos ayudó a desarrollar un pensamiento diferente enfocado en nuestra propia creatividad para resolver problemas, en vez de memorizar fórmulas".
- "Se debería de ampliar el plan de estudios y enseñar conocimientos prácticos de programación no solo la teoría básica, si no, el cómo usarla realmente".
- "Me gustó que la dinámica fuera hacer un juego porque así aprendemos y no se hace tan tediosa la materia".
- "Siento que el curso fue muy didáctico y mejorando las formas de enseñanza".

Revista EIA https://doi.org/10.24050/reia.v22i44.1826

Tabla 2. *Aspectos evaluados en el proyecto final del juego serio (lista de cotejo).*

Calif	Descripción del criterio a evaluar
+5	Calculó y mostró reto correctamente: tiempo y ditancia aleatorio (rango valores válidos)
+5	Calculó posición aleatoria del barril al inicio de cada jugada (rango posiciones válidas)
+5	Calculó y mapeó distancia a coordenadas de pantalla para indicar con flecha la distancia correcta
+5	Modifica tiempo máximo de juego para cada nivel (*timer*)
+5	Incorporó cambio de escena al agotarse tiempo y mostró ventana con pregunta
+5	Mostró nivel y marcador correctamente al cambiar de nivel y al final (variables globales)
+5	Incorporó nuevas imagenes del personaje y el bote (escala 1:1)
+5	Incorporó nuevas imágenes de fondo para cada pantalla/nivel (1024 x 600)
+5	Incorporó font de texto interesante, nítido y original
+5	Incorporó música de fondo para cada pantalla/nivel
+5	Incorporó efectos de sonido (*click*, caída, colisión, enceste, fin-nivel)
+10	Desarrolló nivel 2 a partir del ejemplo del nivel 1 cambiando reto=distancia aleatoria
+5	Incorporó pantalla inicial con datos completos del juego, institución, materia y estudiante
+5	Incorporó pantalla final con imagen de fondo, botón y puntuación acumulada
+5	Agregó al menos una pregunta nueva en cada nivel de juego
+5	Grabó video del juego con duración máx. 2 mins. incluyendo todas las escenas
+5	Participó por lo menos en dos cirugías en clase
+5	Elaboró reporte final completo conforme a la rúbrica correspondiente
+5	Documentó pantallas del juego en el reporte y explicó programas realizados
+5	[Opcional] Diseñó conjunto temático estetico y uniforme (imágenes y sonidos incorporados)
+5	[Opcional] Recopiló y reportó enlaces a los recursos obtenidos en internet (*assests*)
+5	[Opcional] Presentó avances y proyecto final ante el grupo
+5	[Opcional] Incluyó animación o efecto especial: *tweens*, partículas, explosión, nubes, humo, etc.
+5	[Opcional] Elaboró un diseño de imagen, animación o sonido original

5. Discusión y Trabajo Futuro

Comparando la presente propuesta con trabajos similares (Tabla 3), se encuentra que: 1) Jácome et al. (2021) usa gamificación para el desarrollo de habilidades matemáticas en niños sin involucrar el pensamiento computacional, mentoría o retos desvanecidos; 2) Clark y Sengupta (2019) presentan Gameblox, similar a Scratch,

para desarrollar juegos serios para Ciencias integrando habilidades del pensamiento computacional (matematización y simulación), sin aplicar aprendizaje cognitivo supervisado, ni retos desvanecidos; 3) Widiningrum et al. (2024) presentan un trabajo donde los alumnos resuelven problemas a partir de un folleto de Física, creando simulaciones sencillas mediante Scratch, sin embargo, no aplica ningún método del aprendiz cognitivo supervisado; 4) Spieler et al. (2020) aborda el uso de ejemplos trabajados para el desarrollo de habilidades en la resolución de problemas de programación usando juegos serios, sin embargo, no incorpora ningún esquema de aprendizaje supervisado basado en retos ni la matematización o simulación de fenómenos físicos; 5) Van der Linden et al. (2024) ofrece un juego serio para facilitar la comprensión de la Física Newtoniana de una forma lúdica, sin embargo carece de un esquema de supervisión gradual y tampoco incorpora habilidades del pensamiento computacional.

Tabla 3. *Tabla comparativa de trabajos con juegos serios y estrategias de aprendizaje activo.*

No	TRABAJO	Area	Niv. Educ.	PBL	CT	CLT	BT	FWE	LC	SG
1	Jácome et al (2021)	Matem.	Básico			✓	✓			Jugar
2	Clark & Sengupta (2019)	Ciencias	Medio	✓	✓					**Crear**
3	Widiningrum et al (2024)	**Física**	Medio	✓	✓			✓		**Crear**
4	Spieler et al (2020)	Ciencias	Medio	✓		✓		✓		Jugar
5	Van der Linden et al (2024)	**Física**	Básico/ Med	✓						Jugar
6	Método propuesto (2024)	**Física**	Superior	✓	✓	✓	✓	✓	✓	**Crear**

PBL=Problem-based learning. CT=Computational Thinking. CLT=Cognitive Load Theory.

BT=Bloom's taxonomy. FWE=Faded Worked examples. LC=Live Coding. SG=Serious Games.

Entre otros hallazgos destacables se encuentran: 1) este método requiere un esfuerzo inicial significativo tanto por parte del mentor como del aprendiz; 2) las cirugías se aproximan en gran medida al proceso real de programación experta describiendo además las habilidades del pensamiento computacional aplicadas en un proceso

Revista EIA https://doi.org/10.24050/reia.v22i44.1826

de aprendizaje más dinámico, colaborativo, personalizado y tolerante a los errores, mismos que son retro-alimentados y corregidos en tiempo real por el mentor (ejemplos de cirugías disponibles en YouTube: https://youtu.be/ehtwI1lUfgI; https://youtu.be/uDZzhLY_BVo); 3) a pesar de existir cierta incertidumbre al inicio del curso, la encuesta diagnóstica (In_1) evidenció cambios de actitud y motivación favorables, como también lo confirmaron las habilidades adquiridas las subsecuentes sesiones de programación en vivo. Al aumentar las "cirugías" su grado de dificultad, fue palpable una mayor autonomía y nivel de competencia del aprendiz para resolver problemas con la asistencia del mentor, el uso didáctico de los retos desvanecidos y la programación en vivo.

Entre las limitaciones del método propuesto se encuentran: 1) al igual que el método de Halsted, los aprendices menos capaces tienden a rezagarse; 2) método individualizado orientado que requiere una continua supervisión personalizada; 3) la curva de aprendizaje del motor de juegos Godot es más prolongada que otras herramientas como Scratch o Geogebra, pero menor a otros entornos como Unity o Unreal Engine; 4) se requiere que cada aprendiz use un equipo de cómputo apropiado para crear juegos; 5) requiere tiempo y una planificación detallada (Tabla 1); 6) a diferencia del método tradicional, es necesario que todos los aprendices demuestren en vivo sus competencias; 7) el método implica un mentor experimentado, comprometido, positivo y empático (Rowley, 1999).

Asimismo, cabe destacar al integrar el pensamiento computacional, desarrollo de juegos serios y la programación en vivo, será interesante explorar si este enfoque es aplicable para otras asignaturas tipo STEM. A futuro será necesario refinar el método propuesto y mejorar el proceso de seguimiento, así como realizar pruebas de forma colegiada con más grupos evaluando de manera más formal y cuantitativa las competencias involucradas.

6. Conclusiones

El presente trabajo aporta una novedosa metodología relacionada con la aplicación del pensamiento computacional a otras disciplinas, sustentada en las definiciones $C_{1\text{-}8}$ como marco teórico, integrando

diversos métodos y estrategias de aprendizaje activo supervisado con retos desvanecidos, denominado **método del cirujano con retos devanecidos**, el cual se organizó en tres etapas para facilitar el proceso gradual de mentoría, supervisión y aprendizaje. Cotejando las preguntas de investigación, en basde a los resultados parciales obtenidos por medio de los instrumentos In_{1-4}, podemos concluir que:

- RQ_{1-3}. Las respuestas de los propios estudiantes participantes en estos instrumentos reportan de forma mayoritaria un alto grado de aceptación, motivación y actitud positiva, proactiva, autónoma y autogestiva. A pesar de que la mitad de los estudiantes cursaban por segunda ocasión el curso de Física y/o no sabían programar, todos los participantes concluyeron con éxito el desarrollo del videojuego apoyándose del mentor, de sus compañeros, consultando videos y asistentes virtuales (chatbots).
- RQ_4. La exploración experimental mostró que a pesar de contar con una exigente curva de aprendizaje y una laboriosa planificación, este instrumento y las intervenciones (cirugías), evidenciaron una experiencia didáctica orgánica y dinámica, propiciando una sana competencia basada en la superación, inclusión y diversidad de perspectivas, superación gradual, tolerancia al error, colaboración, gran participación y un aprendizaje activo, significativo, personalizado, crítico, pragmático, creativo, lúdico y autogestivo.

7. Agradecimientos

Al Tecnológico Nacional de México, la academia y los grupos de Ciencias Básicas del Tecnológico de Chihuahua, la UACH y los colegas que revisaron el manuscrito original.

8. Referencias

ACM. (2023). *Bloom's for Computing: Enhancing Bloom's Revised Taxonomy with Verbs for Computing Discipline*. *ACM Comm. Comp. Educ. Community Colleges (CCECC)*. Recuperado de https://ccecc.acm.org

Altun, S., & Morkoyunlu, Z. (2023). The use of error based activities to improve the mathematization competency. *International Journal of Progressive Education*, *19*(3), 134–148. https://doi.org/10.29329/ijpe.2023.546.8

Aondofa, A., & Atindi, V. (2022). Effectiveness of the use of gamification by teachers of physics in secondary schools in Buruku Benue State. *International Journal of Academic Multidisciplinary Research*, *6*(8), 6–11.

Aprilia, A., et al. (2023). Attitudes and beliefs toward physics: A mini review. *Revista Mexicana de Física*, *E 20*(1), 1–8.

Azzouzi, A., et al. (2022). Physics mathematization: Teachers' observations on the application of ICT. *International Conference on Intelligent Systems and Computer Vision (ISCV)*, 1–5. https://doi.org/10.1109/ISCV54655.2022.9806103

Ben Rebah, H., & Ben Slama, R. (2019). The educational effectiveness of serious games. *Mediations and Mediatizations*, *2*(1), 131–155. https://doi.org/10.52358/mm.vi2.97

Bilbao, J., et al. (2024). Introducing computational thinking and algebraic thinking in the European educational systems. *International Journal of Education and Information Technologies*, *18*, 11–19.

Chen, O., et al. (2023). The effect of worked examples on learning solution steps and knowledge transfer. *Educational Psychology*. https://doi.org/10.1080/01443410.2023.2273762

Chi, M., et al. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, *13*(1), 145–182.

Clark, D., & Sengupta, P. (2019). Reconceptualizing games for integrating computational thinking and science as practice. *Interactive Learning Environments*, *28*(3), 328–346. https://doi.org/10.1080/10494820.2019.1636071

Clark, R., Nguyen, F., & Sweller, J. (2006). *Efficiency in learning: Evidence-based guidelines to manage cognitive load*. Pfeiffer Wiley.

Collins, A., et al. (1987). *Cognitive apprenticeship*. Technical Report No. 403.

Espinal, A., et al. (2024). Professional development in computational thinking: A systematic literature review. *ACM Transactions on Computing Education*, *24*(2), 27. https://doi.org/10.1145/3648477

Furuta, T., & Knezek, G. (2023). Text mining analyses of programming education articles since the 1970s. *Information and Technology in Education and Learning*, *3*(1). https://doi.org/10.12937/itel.3.1.Reg.p001

Gazis, A., & Katsiri, E. (2023). Serious games in digital gaming: A comprehensive review of applications. *WSEAS Transactions on Computer Research*, *11*, 10–22. https://doi.org/10.37394/232018.2023.11.2

Godot. (2024). *Motor de juegos de código abierto Godot*. Recuperado de https://godotengine.org

Gurbuz, S., & Celik, M. (2022). Serious games in future skills development: A systematic review of the design approaches. *Computer Applications in Engineering Education*, *30*(5), 1591–1612. https://doi.org/10.1002/cae.22557

Hake, R. (1998). Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics*, *66*(1), 64–74. https://doi.org/10.1119/1.18809

Hamidi, A., et al. (2023). A complementary view to computational thinking and its interplay with systems thinking. *Educational Sciences*, *13*(2), 201. https://doi.org/10.3390/educsci13020201

Hestenes, D. (1992). Modeling games in the Newtonian world. *American Journal of Physics*, *60*(8), 732–748.

Hidayat, R., et al. (2022). Achievement goals, metacognition and horizontal mathematization: A mediational analysis. *TEM Journal*, *11*(4), 1537–1546. https://doi.org/10.18421/TEM114-14

Ivanjek, L., et al. (2024). Enhancing mathematization in physics education by digital tools. *Physics Education Today*, 35–53. Springer Nature.

Jácome, L., et al. (2021). Gamification as an educational strategy to strengthen cognitive abilities of mathematics in school children. *Advances in Intelligent Systems and Computing*, *1277*, 142–150. https://doi.org/10.1007/978-3-030-60467-7_12

Kılıç, S. (2022). Tendencies towards computational thinking: A content analysis study. *Participatory Educational Research*, *9*(5), 288–304. https://doi.org/10.17275/per.22.115.9.5

Kotsis, S., & Chung, K. (2013). Application of the "See One, Do One, Teach One" concept in surgical training. *Plastic and Reconstructive Surgery*, *131*(5), 1194–1201. https://doi.org/10.1097/PRS.0b013e318287a0b3

Lukman, H., et al. (2023). Gamification of mathematics teaching materials: Its validity, practicality and effectiveness. *International Journal of Emerging Technologies in Learning (iJET)*, *18*(20), 4–22. https://doi.org/10.3991/ijet.v18i20.36189

Marusic, M., & Slisko, J. (2012). Increasing the attractiveness of school physics: The effects of two different designs of physics learning. *Revista Mexicana de Física*, *E 58*(1), 75–83.

Ngadengon, Z., et al. (2024). Theory on computational thinking in education: A systematic review. *International Journal of Education, Psychology and Counseling*, *9*(53), 488–507.

Pacheco, A. (2024). Juegos serios: Matematizando la física de forma lúdica. *Electro*, *arXiv* [por publicar]. https://doi.org/10.48550/arXiv.2407.10057

Prince, M., & Felder, R. (2006). Inductive teaching and learning methods: Definitions, comparisons, and research bases. *Journal of Engineering Education*, *95*(2), 123–138. https://doi.org/10.1002/j.2168-9830.2006.tb00884.x

Rowley, J. (1999). The good mentor. *Educational Leadership*, *56*(1), 20–22.

Spieler, B., et al. (2020). Reducing cognitive load through the worked example effect within a serious game environment. *IEEE International Conference on Interactive Learning Environments (iLRN)*, 1–8. https://doi.org/10.23919/iLRN47897.2020.9155187

Tomalá, J., et al. (2020). Serious games: Review of methodologies and game engines for their development. *Iberian Conference on Information Systems and Technologies (CISTI)*, 1–6. https://doi.org/10.23919/CISTI49556.2020.9140827

Van der Linden, A., et al. (2024). Learning Newtonian mechanics with an intrinsically integrated educational game. *Journal of Computer Assisted Learning*, 1–11. https://doi.org/10.1111/jcal.12966

Van Hemmen, J. (2021). Mathematization of nature: How it is done. *Biological Cybernetics*, *115*, 655–664.

Vorhölter, K. (2021). Metacognition in mathematical modeling: The connection between metacognitive individual strategies, metacognitive group strategies and modeling competencies. *Mathematical Thinking and Learning*, *25*(3), 317–334. https://doi.org/10.1080/10986065.2021.2012740

Wells, M., Hestenes, D., & Swackhamer, G. (1995). A modeling method for high school physics instruction. *American Journal of Physics*, *63*(7), 606–619.

Widiningrum, W., et al. (2024). Effect of PBL-based Scratch e-module in improving computational thinking and physics concepts. *International Journal on Studies in Education and Science (IJSES)*, *5*(2), 124–139. https://doi.org/10.46328/ijres.76

Wilson, J., & Clarke, D. (2003). Towards the modelling of mathematical metacognition. *Mathematics Education Research Journal*, *16*(2), 25–48. https://doi.org/10.1007/BF03217394

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