

Microaprendizaje con recursos desarrollados en Canva para la enseñanza de conceptos de programación usando GameMaker en estudiantes de la CIDeV

Microlearning with resources developed in Canva for teaching programming concepts in GameMaker with CIDeV students

Adriana Lucía Céspedes Vindas¹

Resumen

Introducción: El fenómeno de la matemafobia, descrito por Seymour Papert, también se manifiesta en la programación, donde muchas personas experimentan temor o sensación de dificultad ante el aprendizaje de las matemáticas. En la Comunidad Interuniversitaria de Desarrolladores de Videojuegos (CIDeV) de Costa Rica, se busca que estudiantes de áreas no computacionales adquieran habilidades básicas en motores de programación para crear prototipos de videojuegos viables y venzan la preconcepción de que la programación es difícil y aburrida. Sin embargo, las capacitaciones existentes carecen de una adecuada mediación didáctica, lo que limita su efectividad. **Objetivo:** Este artículo tiene como objetivo evaluar una estrategia didáctica basada en módulos interactivos y microaprendizaje para fortalecer la comprensión de los conceptos fundamentales de programación en el contexto del desarrollo de videojuegos dentro de actividades en la CIDeV. **Metodología:** Mediante la metodología IBD (Investigación Basada en el Diseño) se elaboraron cuatro módulos sobre algoritmos, variables, objetos, condicionales y funciones, integrando tutoriales prácticos con el motor Game Maker. Participaron 24 estudiantes, cuyos conocimientos fueron evaluados antes y después de la intervención. **Hallazgos:** Los resultados evidencian una mejora general, más equilibrada y de mejor desempeño. Además, en un taller presencial se observó la aplicación creativa de los conceptos aprendidos, lo que reflejó una mejor comprensión y apropiación. **Conclusiones:** Se concluye que la afinidad hacia los videojuegos influyó positivamente en la disposición a explorar herramientas de programación, mitigando el temor inicial y promoviendo una actitud más abierta hacia el aprendizaje técnico en estudiantes de la CIDeV.

Palabras Clave: Construccinismo, Microaprendizaje, Programación, Videojuegos

¹ Adriana Céspedes Vindas es bachiller en Ciencias de la Computación de la Universidad de Costa Rica, Licenciada en Ingeniería Informática con énfasis en Administración de Proyectos de la Universidad Estatal a Distancia. Actualmente, maestrante del Programa de Maestría en Tecnología e Innovación Educativa (MATIE) de la Universidad Nacional. Trabaja en el Laboratorio de Investigación e Innovación Tecnológica (LIIT) de la UNED como analista académico con intereses en Ciencia de Datos, Género en STEAM y Geoinformación.

Abstract

Introduction: The phenomenon of math phobia, described by Seymour Papert, also manifests in programming, where many people experience fear or a sense of difficulty when learning mathematics. In the Interuniversity Community of Videogame Developers (CIDEV) of Costa Rica, the goal is for students from non-computing fields to acquire basic skills in programming engines in order to create viable video game prototypes and overcome the preconceived notion that programming is difficult and boring. However, the existing training programs lack adequate didactic mediation, limiting their effectiveness.

Objective: This article aims to evaluate a didactic strategy based on interactive modules and microcontent to strengthen the understanding of fundamental programming concepts in the context of video game development. **Methodology:** Using Design-Based Research (DBR) methodology, four modules were developed on algorithms, variables, objects, conditionals, and functions, integrating practical tutorials with the Game Maker engine. A total of 24 students participated, and their knowledge was assessed before and after the intervention. **Findings:** The results show overall improvement, with more balanced performance and a reduction in low scores. Furthermore, during an in-person workshop, the creative application of the learned concepts was observed, reflecting better understanding and ownership. **Conclusions:** It is concluded that affinity for video games positively influenced the willingness to explore programming tools, mitigating initial fear and promoting a more open attitude toward technical learning among CIDEV students.

Keywords: Constructionism, Microlearning, Programming, Video Games

Introducción

En los espacios educativos, la programación muchas veces es vista como un proceso complejo y difícil, dado que tiene una fuerte asociación con la matemática. Las personas estudiantes tienen que superar muchas veces una barrera de miedo para animarse a programar. Existe un miedo y resistencia a codificar, que se suma a una alta demanda cognitiva que se requiere para resolver problemas y comprender conceptos como la abstracción, mientras se aprende un nuevo lenguaje, con su propia sintaxis y escritura. Esta situación lleva a muchos estudiantes a desertar en el proceso. (Osman, 2020; Alamina et al., 2025; Shakib et al., 2025).

La Comunidad Interuniversitaria de Desarrolladores de Videojuegos (en adelante, CIDEV) es un grupo de interés especial conformado por estudiantes y docentes universitarios costarricenses. Los profesionales y estudiantes que participan provienen de distintas carreras, principalmente del área de las artes, la música y la ingeniería computacional, y cuentan con distinto nivel académico. Uno de los propósitos principales de la CIDEV es acercar a personas jóvenes a la industria y al

emprendimiento a través de actividades formativas como talleres y maratónicas de improvisación llamadas “Game Jams” (Mora-Zamora y Céspedes-Vindas, 2024).

Durante las maratónicas y talleres de desarrollo de videojuegos que organiza la CIDeV en Costa Rica, se ha percibido que en algunos equipos de trabajo no se cuenta con personas con experiencia en programación. Esto coloca a los integrantes en una posición difícil con respecto a la toma de decisiones acerca del uso de las herramientas de desarrollo y lenguajes de programación en las actividades, condicionando el alcance de las creaciones. En estas situaciones, es común que personas del área de artes tomen el rol de programadoras. De esta forma, para responder a los objetivos de aprendizaje de la CIDeV en la formación de personas en programación, se percibe que se requieren recursos, capacitaciones o guías de nivelación que permitan a quienes participan adentrarse en el mundo de la programación de manera más accesible y adaptada a sus necesidades, de forma que puedan participar de manera más activa en sus espacios.

Aunado a lo anterior y en el marco de una de las maratónicas organizadas, llamada *Women Game Jam*², la CIDeV desarrolló videotutoriales dirigidos a público principiante sobre el uso de las herramientas de software: *Unity*, *Unreal Engine*, *RPG Maker*, *Game Maker* y *RenPy*. Sin embargo, los tutoriales “no han sido efectivos, a excepción del de Renpy que sí tenía una mediación didáctica clara” (R. Mora, Comunicación personal, 16 Septiembre 2025). La mayoría de estos recursos han sido desarrollados de forma apresurada y se percibe que no logran uno de los principales objetivos de la CIDeV, que es acercar a las personas a los engines de programación: “El engine es una herramienta integradora y es indispensable que todos los roles técnicos sepan usarlo, aunque sea superficialmente, por esto tener experiencias accesibles que acerquen a estos perfiles al engine es sumamente valioso” (R. Mora, Comunicación personal, 16 Septiembre 2025).

Por lo tanto, es de especial interés para la CIDeV crear productos didácticos que no solo les permitan a las personas participantes completar un juego como parte de sus actividades, sino que les den las herramientas y confianza para crear sus propios productos, superando el miedo a programar. Sobre esto, para Seymour Papert

² La Women Game Jam es una maratónica de desarrollo de videojuegos de carácter internacional dirigida a mujeres cis, trans y personas no binarias (Céspedes-Vindas y Mora-Zamora, 2024).

(1984), el aprendizaje surge de la construcción activa y del vínculo personal con lo que se aprende. Como señala: “Piaget habla casi exclusivamente de los aspectos cognoscitivos de la asimilación. Pero hay también un componente afectivo” (Papert, 1984, p. 12). Es decir, en espacios como los que promueve el CIDEV, no basta con transmitir contenidos, por lo que se necesita emoción, curiosidad y espacios para experimentar con la programación que pueden influir de manera positiva en el aprendizaje.

Enseñar a programar requiere crear un ambiente que favorezca la adquisición de conocimientos y conceptos, que vayan acompañados de ejercicios de implementación. Para Wing (2006), “pensar como un computólogo significa más que ser capaz de programar una computadora. Requiere pensar en múltiples niveles de abstracción” (p. 33). Es decir, Wing (2006) recalca que es precisamente ese pensamiento computacional, el que va más allá, consiste en habilidades esenciales para la vida en la sociedad actual, que pueden descubrirse y fortalecerse a través de diversas actividades, útiles no solo para informáticos sino para diversas áreas profesionales.

En este marco, los componentes lúdicos de los juegos surgen como una opción que puede promover la motivación por aprender a programar. Como mencionan Sandí y Sanz (2018), permiten atraer poblaciones distintas al aprendizaje, haciendo la experiencia divertida. Es así como, crear y diseñar un juego desde cero a través de un lenguaje de programación es una técnica bastante utilizada para aprender. Un ejemplo es la opción del entorno de juegos Game Maker (Acosta et al., 2020; AYYILDIZ, 2020; Gajewski et al., 2022), que provee opciones dentro del motor para programar, con interfaces más alejadas de la sintaxis a través de un lenguaje gráfico, o usando GML³ (Game Maker Language), un lenguaje propio de Game Maker, más cercano a una sintaxis de lenguajes como Java o Javascript.

La comunidad CIDEV trabaja como un entorno STEAM transdisciplinar (Mora-Zamora y Céspedes-Vindas, 2024) y busca crear capacidades en los estudiantes de forma tal que se pierda el miedo a probar nuevas disciplinas, favoreciendo la

³ Game Maker Language es un lenguaje de programación propio del software *Game Maker*. Es un tipo de lenguaje imperativo con sintaxis similar a lenguajes de programación de C ó Javascript.

adquisición de competencias valiosas para el desarrollo integral, principalmente con engines de juegos. Actualmente, la comunidad tiene interés en atraer más estudiantes de las áreas de arte, música, animación, muchos con conocimientos nulos o muy básicos en programación. En este escenario particular, se hace relevante el objetivo de este trabajo: *Desarrollar una estrategia didáctica de módulos interactivos basada en microcontenidos de programación de videojuegos para mejorar el aprendizaje de los conceptos fundamentales de programación en estudiantes del CIDeV.*

En este contexto, la estrategia se entiende como el conjunto de acciones, técnicas y recursos utilizados para facilitar el aprendizaje de los conceptos de programación. Así, se busca medir los aportes que puede dar a la comunidad un producto educativo como el que se plantea, evaluando los resultados y mejora de las bases de conocimientos de los estudiantes que participan de las actividades organizadas por la comunidad.

Marco Conceptual

Construccionismo y Constructivismo para “aprender haciendo”

En los paradigmas emergentes que promueven el desarrollo del pensamiento computacional, se reconoce que el aprendizaje no debe limitarse a lo intelectual, sino que involucra también la importancia de los procesos sobre los resultados y las dimensiones creativas, sociales y culturales (Rodríguez et al., 2018; Nieves y Pérez, 2021). El construccionismo, impulsado por Papert, sostiene que el conocimiento surge cuando los individuos producen artefactos significativos, digitales o físicos, que median entre su pensamiento y el mundo (Lodi y Martini, 2021). Este énfasis en la creación resulta especialmente pertinente en la enseñanza de programación mediante proyectos de videojuegos, pues permite a los estudiantes de arte y música vincular los conceptos técnicos con procesos expresivos y la creación de productos creativos propios.

En diálogo con el construccionismo, el constructivismo de Piaget resalta la importancia de la actividad del sujeto en la construcción de su propio conocimiento, al reorganizar y adaptar sus estructuras mentales propias mediante la experiencia para la resolución de problemas (Olmedo, Farrerons y Pujol, 2021; Centro de Investigación y Desarrollo, 2024). Es decir, la persona estudiante aprende a resolver los problemas que

se le presentan, usando las herramientas y conocimientos que ya tenía a disposición. A su vez, Ausubel (Givi et al., 2021; Capecchi et al., 2022) enfatiza el valor del aprendizaje significativo, el cual depende de que los contenidos sean relevantes e interesantes para el aprendiz. De este modo, tanto el constructivismo como el construccionismo justifican uno de los propósitos de la CIDEV: enseñar programación a través de la creación de videojuegos, en un marco de aprender haciendo que combine la experiencia activa con la motivación de un producto lúdico atractivo, sentando la base para la estrategia didáctica de los módulos propuestos.

El miedo en el proceso de aprender a programar

El pensamiento computacional implica la reformulación de problemas, su descomposición o división en partes más pequeñas, el modelado con representaciones (abstracción) y el diseño de pruebas y acciones para encontrar las soluciones (Wing, 2006). Es un pensamiento estructurado y algorítmico que busca no solo resolver el problema sino también producir resultados, donde no es solo importante incluir el diseño sino también el entendimiento del comportamiento de las variables (Angeli y Giannakos, 2020). Este enfoque puede percibirse como ajeno por parte de estudiantes sin formación en ciencias exactas, generando resistencia o inseguridad hacia su aprendizaje como se ha visto con un porcentaje importante de los participantes de la CIDEV.

Al trasladar esta reflexión a la programación, el problema no radica en la dificultad intrínseca del lenguaje formal de programación, sino en la manera en que se introduce. Papert (1984) propone tres principios para contrarrestar la matemafobia o aversión hacia la matemática, que resultan igualmente válidos para la programación: continuidad, al conectar los nuevos aprendizajes con experiencias previas cargadas de sentido; poder, al permitir que los estudiantes utilicen lo aprendido en proyectos significativos; y resonancia cultural, al situar los contenidos en contextos socialmente relevantes.

En la CIDEV, esos principios se concretan en proyectos de videojuegos, donde la programación deja de ser un fin en sí mismo y se convierte en un medio para construir narrativas, personajes y productos interactivos. Dentro de este contexto, herramientas

como la programación por bloques, metáforas visuales y entornos con interfaces gráficas fáciles de usar y atractivas han logrado acercar el pensamiento computacional a estudiantes (Pérez-Marín et al., 2020). Así, el reto no es solo enseñar estructuras básicas como condicionales, ciclos o funciones (Matthes, 2023), sino lograr que los estudiantes reconozcan en ellas un recurso expresivo al servicio del diseño de sus propios juegos.

El microaprendizaje y los tutoriales en el aprendizaje

El microaprendizaje ha surgido como una estrategia pedagógica informal, flexible y accesible, caracterizada por fragmentar los contenidos en unidades pequeñas y de rápida asimilación, en un tipo de aprendizaje de “pequeños pasos” (Thillainadesan et al., 2022; Fidan, 2023; Zamata-Aguirre et al., 2023). Su efectividad radica en que permite reducir la carga cognitiva y adaptarse a los ritmos de aprendizaje de los estudiantes, aspecto especialmente útil para quienes se enfrentan a la programación por primera vez.

En el caso de la CIDEV, el microaprendizaje se vincula con la enseñanza de programación a través de contenidos teóricos y prácticos que llevan a la construcción de un videojuego. En este sentido, los tutoriales en formato de video o multimedia se consolidan como recursos complementarios de fácil aprovechamiento que potencian el aprendizaje autónomo y que promueven el aprendizaje reflexivo (Gascho y Slebodnik, 2024; Rodríguez, 2020). No obstante, como advierten Lewis, Lewis y Luca (2023), no basta con producir videos; estos deben estar guiados por objetivos claros y alineados con la experiencia de aprendizaje.

Es importante resaltar que se ha visto que estudiantes con baja comprensión lectora se benefician de recursos multimedia pero en algunos casos los blogs de texto son mejores que los videos para la comprensión (Delgado, Anmarkrud, y Ávila, 2022). De ahí la importancia de la diversidad de formatos y las estrategias que se usan para presentar el contenido. Así, la combinación de microlearning y tutoriales puede entenderse como una extensión práctica de los principios constructivistas y construccionistas, dado que promueven el aprendizaje significativo mediante

contenidos breves y personalizados, mientras se refuerza la autonomía del estudiante y su capacidad de vincular la programación con intereses artísticos.

Los videojuegos, más allá que un producto lúdico

El uso de videojuegos en la educación ha demostrado ser una estrategia eficaz para enseñar tanto habilidades de programación como conceptos de la ciencia computacional (Holenko y Hoic-Bozic, 2021). Esto se manifiesta en que la participación de los estudiantes en la creación o modificación de juegos no solo estimula la práctica de estructuras de programación, sino que también les ofrece un espacio lúdico y motivador donde se tiene un camino o desafío de conocimiento, sin importar el resultado, siendo la construcción una de las mejores formas de aprender (Rugelj y Lapina, 2019).

Aunque los videojuegos suelen asociarse únicamente con entretenimiento, investigaciones recientes destacan su potencial para enseñar y educar. En varios estudios educativos donde hay un test al inicio y otro al final, la mayoría de las personas que han estado en contacto con un videojuego han visto incrementada su habilidad cognitiva como la toma de decisiones o la gestión del tiempo (Reynaldo et al., 2021). En la CIDEV, los videojuegos cumplen una doble función: son el producto central de los proyectos y, al mismo tiempo, el medio pedagógico que permite aprender haciendo.

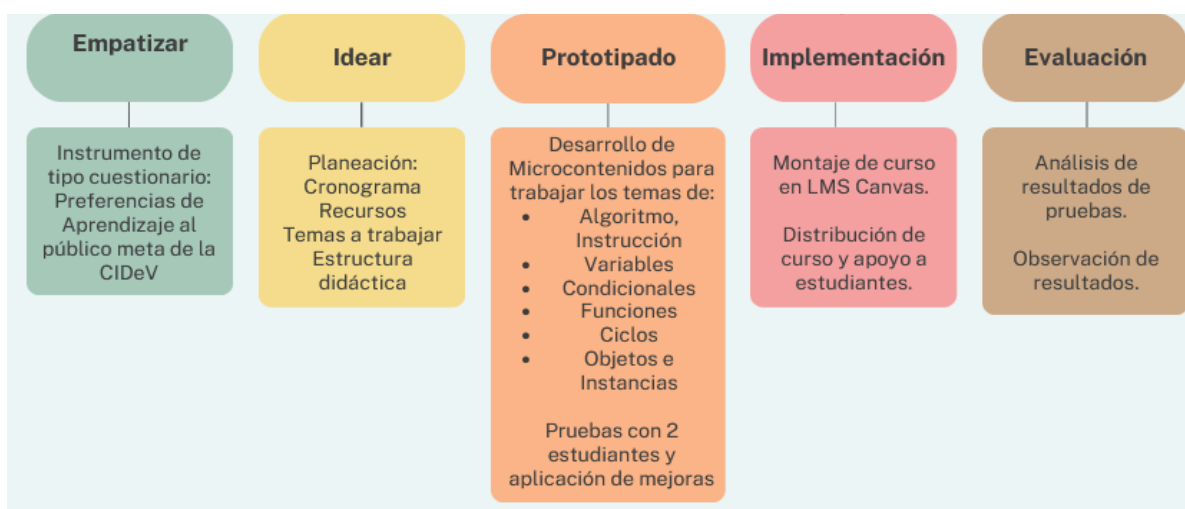
A través de los videojuegos, la programación se transforma en una experiencia significativa y creativa, accesible para estudiantes que provienen de disciplinas tradicionalmente alejadas de la ciencia computacional o la programación. Además, experiencias previas como las de (Acosta et al., 2020; AYYILDIZ, 2020; Gajewski et al., 2022) con el uso del motor de juego Game Maker muestran buenos resultados de enseñanza y aprendizaje, donde la herramienta es una buena vía de entrada para acercar a estudiantes a lenguajes con sintaxis de programación a través del diseño.

Metodología

Este estudio se desarrolló con un enfoque cualitativo, que permite analizar los complejos y diversos procesos de aprendizaje individuales de los estudiantes en el acercamiento a la programación.

Para ello se implementó una estrategia de Investigación Basada en Diseño (IBD), que permite responder directamente a las necesidades del público meta y buscar la generación de conocimiento en colaboración con las personas estudiantes. En la Figura 1, se muestran las actividades realizadas por cada etapa metodológica.

Figura 1. Metodología aplicada a proyecto



Nota: Elaboración propia

Etapa “Empatizar”: En esta fase, se distribuyó un instrumento de tipo cuestionario a estudiantes de la CIDEV a través del canal de WhatsApp de la comunidad. A los estudiantes se les preguntó información demográfica como la edad y género. También, se les consultó por los medios más utilizados para aprender herramientas de software, incluyendo un espacio abierto para que los estudiantes compartieran recursos de aprendizaje en video que les han resultado útiles para aprender en el pasado. Durante esta etapa, se notó una marcada preferencia por los contenidos en video para aprender.

Etapa “Idear”: Con base a la preferencias obtenidas de la etapa anterior, se optó por desarrollar una serie de microcontenidos en video para explicar los conceptos

de programación y la funcionalidad del Engine de programación. Cada microcontenido se acompañó de un texto corto con imágenes, dado que fue el segundo recurso planteado por los estudiantes como fácil de consumir y amigable con ellos para aprender herramientas de software. Para desarrollar las actividades prácticas se seleccionó enseñar al estudiantado el uso del motor de desarrollo de videojuegos Game Maker. Este software fue seleccionado por su compatibilidad con sistemas operativos, dado que no es tan pesado y cuenta con una interfaz gráfica amigable con lenguaje de cuarta generación (cajas lógicas), que lo hace muy amigable respecto a otras alternativas como Unity o Unreal donde sí se requieren bases de programación de entrada. La herramienta también ofrece un lenguaje de scripting llamado GML (Figura 2). Los ejemplos trabajados para los microcontenidos fueron desarrollados con GML para permitir que los estudiantes estuvieran más cerca de la sintaxis de un lenguaje de programación de scripting como JavaScript.

Figura 2. Ejemplo de Script de código en GML

```
13 if (place_meeting(x, y, maincharacter)){
14     global.lives = global.lives-1;
15 }
```

Nota:elaboración propia

Juego a desarrollar y conceptos para aprender: El trabajo didáctico de los microcontenidos gira en torno a enseñar a las personas participantes cómo construir un videojuego en Game Maker, el cual como motor se adapta muy bien a juegos de plataformas 2D. Este estilo de videojuegos, aparte de ser familiar debido a clásicos como *Super Mario*, también permite trabajar temas como: Objetos, Eventos, Condicionales y Variables de forma natural, ya que cada colisión con un elemento de juego (sea enemigo, premio o plataforma) requiere la implementación de estos conceptos a través de la programación algorítmica de la solución.

Etapas “Prototipado” - Microcontenidos desarrollados : Con base en los insumos planteados y la estrategia didáctica propuesta, se desarrollaron 23 microcontenidos en video (Tabla 1). Los microcontenidos son de dos tipos:

- **Fundamentos-Conceptos:** En estos videos se explican los fundamentos teóricos de programación como: algoritmo, instrucción, variable, clase, objetos, funciones, condicionales, a través de metáforas y ejemplos.
- **Prácticos-Tutoriales.** En estos se aplican los conceptos teóricos para construir el juego con apoyo del motor. Son más guiados e instrumentales.

Tabla 1. Títulos de los microcontenidos en video desarrollados con su duración

Micocontenidos Teóricos (Duración mm:ss)	Microcontenidos Prácticos (Duración mm:ss)
1. Algoritmos (2:07) 2. Instrucciones (0:55) 3. Variables (1:47) 4. Variables booleanas (1:57) 5. Variables locales y globales (1:23) 6. Clases e instancias (2:02) 7. Eventos (0:42) 8. Atributos (1:12) 9. Funciones (1:59) 10. Condicionales (2:23) 11. Ciclos (4:26)	1. Explorando el entorno (1:11) 2. Carga de Sprites/Imágenes de proyecto (0:55) 3. Salas y ejes (0:50) 4. Creando el objeto Empty (1:59) 5. Creando el objeto Principal (1:06) 6. Creando el objeto Enemigo (1:07) 7. Creando una Sala/Nivel (0:58) 8. Añadiendo comportamiento a objeto Empty (2:19) 9. Añadiendo comportamiento a objeto Caja de Explosivos (1:57) 10. Añadiendo comportamiento a objeto Caja de Sorpresas (1:55) 11. Añadiendo comportamiento a objeto Personaje Principal (3:29) 12. Añadiendo comportamiento a objeto Enemigo (2:18)

Nota: Elaboración propia

En total, se desarrollaron 23 microcontenidos en video. Cada video fue acompañado por un microcontenido texto pequeño que introduce y clarifica conceptos. Tal y como se muestra en la Figura 3.

En la Figura 4, se identifican los componentes básicos o estructura general que siguen la mayoría de los micro contenidos teóricos desarrollados, que consisten en: Definición del concepto (A), Metáforas (B) y Ejemplos prácticos en pseudocódigo (C, D).

Figura 3. Captura de plataforma Canvas- Módulo 3: Clase y Objeto

Resumiendo: Las clases son plantillas, los objetos o instancias se adhieren o usan esa plantilla.

Otros ejemplos:

Clase Animal

Esta *clase* es como una *idea general* de lo que es un animal. Define las características y comportamientos que todos los animales tienen en común.

- **Atributos:** nombre, edad, especie
- **Métodos:** comer(), dormir()

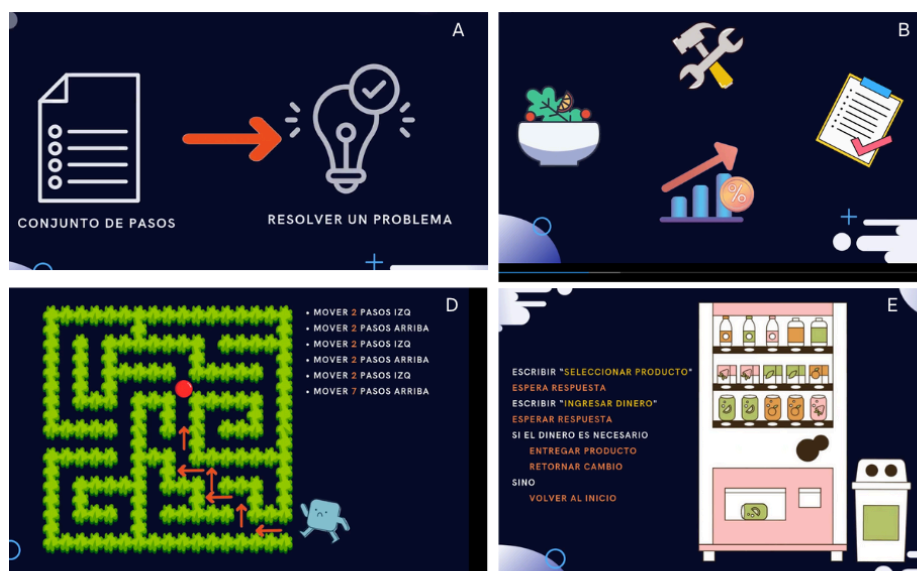
Objetos:

Objeto 1 de tipo Animal:

- nombre: "Firulais"
- edad: 4
- especie: "perro"
- Métodos: comer(), dormir()

Nota: Módulo 3, "Aprendiendo a programar con videojuegos", Plataforma Canvas, Elaboración propia.

Figura 4. Capturas de microcontenido: Módulo 1: Tema Algoritmos



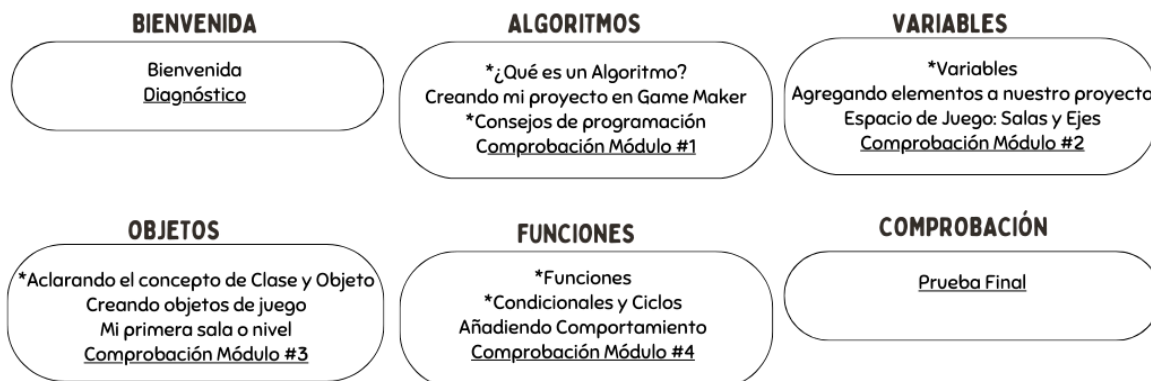
Nota: a) Concepto. b) Metáforas c) Ejemplo algoritmia con laberinto. d) Ejemplifica de algoritmia con una máquina expendedora

Módulo 1, Curso Aprendiendo a programar con videojuegos, Elaboración propia

Plataforma de aprendizaje: La experiencia didáctica fue colocada a disposición de los estudiantes a través de la plataforma LMS (Learning Management System) *Canvas Instructure*. Esto permitió crear módulos de aprendizaje temáticos que mezclaran los microcontenidos teóricos y prácticos siguiendo la estructura de la Figura 5. Al finalizar cada módulo, los estudiantes realizaron una prueba corta de comprobación. El uso de la plataforma permitió llevar el control de las notas y disponer de los recursos y medios de consulta de tipo foro para los estudiantes. Cada prueba

consistía en preguntas de selección única o pareo. Todas las pruebas eran cortas (no más de 10 preguntas por prueba).

Figura 5. Estructura de la plataforma Canvas - Módulos de aprendizaje



Nota: Las páginas con * corresponden a contenido teórico-conceptos. Los ítems subrayados a comprobaciones y los no marcados a tutoriales guiados de uso del motor
Elaboración propia

Etapa “Implementación” : La propuesta didáctica, alojada en Canvas, se distribuyó a grupos de participantes de dos actividades de la CIDeV: un taller de programación dirigido a estudiantes de noveno año del Colegio Técnico Profesional de Alajuelita (CTP Alajuelita) y personas inscritas a la Women Game Jam. Los módulos fueron trabajados de forma asincrónica durante los meses de junio y julio de 2025 por los estudiantes. Posteriormente, durante la última semana de julio 2025, se realizó una sesión presencial de tres horas en laboratorio con ambos grupos, en la cual los participantes desarrollaron sus propios proyectos de juego. En total 24 personas completaron la experiencia (14 estudiantes del CTP Alajuelita y 10 participantes de la WGJ). La totalidad de los 24 participantes declaró no tener conocimientos previos de programación al participar de la experiencia didáctica. Este proyecto se alineó a estas dos actividades de tipo taller que ya tenía calendarizada la CIDeV.

Etapa “Evaluación” : Para esta etapa se consideraron únicamente los registros de los estudiantes que completaron todos los módulos de aprendizaje y la sesión presencial. Se excluyeron a quienes no finalizaron el proceso o participaron de solo una de las actividades.

La adquisición de conocimientos y la comprensión de los conceptos se midieron a partir de la prueba final. Adicional a las notas, se observó la capacidad de resolución de los estudiantes a preguntas abiertas de resolución de problemas. Para esto se solicitó a cada participante seleccionar un problema de programación sencillo, de 4 posibles. Los problemas fueron tomados de la plataforma en línea *BeeCrowd*⁴ (*Juez en línea de programación*). Adicionalmente, se observaron los resultados de la actividad presencial en laboratorio donde los estudiantes aplicaron los conocimientos en proyectos de temática libre en grupo.

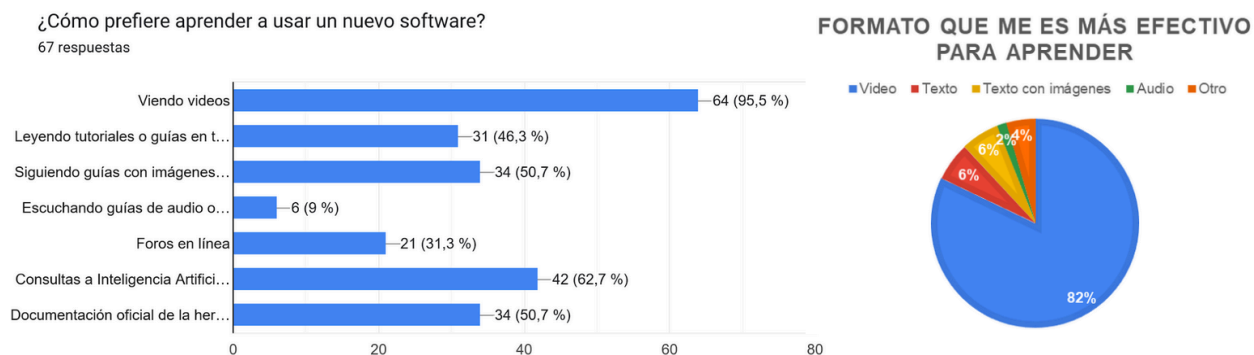
Resultados de la etapa empatía

El cuestionario fue respondido por 67 personas: 33 personas (49%) de 21-25 años, 11 (16%) personas entre 15 y 20 años, 14 (21%) personas entre 26-30 años y 9 personas mayores de 30 años (Figura 2). En cuanto al género, se contó con (48 personas, 71.6%) de género masculino, (15 personas, 22.4%) de género femenino y (4 personas, 6%) de género no binario. Además, más del (41 personas ,61.2%) tienen educación universitaria incompleta y (18 personas, 26.9%) universitaria completa.

En la Figura 6, se muestra el medio favorito para aprender herramientas técnicas, seleccionado por la mayor cantidad de personas de la CIDeV: video (55 respuestas, 82%), texto (4 respuestas, 6%), texto con imágenes (4 respuestas, 6%) y audio (1 respuesta, 2%). Adicionalmente, un dato interesante recopilado es que los estudiantes manifestaron que pese a que la duración preferida se encuentra entre los 5 y 10 minutos (27 personas, 40.3%), en realidad pueden y prefieren videos bien hechos, que sean concisos y claros por sobre la duración.

⁴ BeeCrowd Online Judge es una plataforma con ejercicios de programación competitiva. Tiene recursos disponibles para diferentes lenguajes y maneja distintos niveles de dificultad. Disponible en: <https://judge.beecrowd.com>

Figura 6. Resultados etapa “empatía”: Medios preferidos y recursos preferidos para aprender software



Nota: Elaboración propia

Con la información recopilada en la etapa de empatía se observó que los estudiantes de la CIDeV en su mayoría prefieren los contenidos en formato de video. Por lo tanto, los microcontenidos de la estrategia didáctica fueron trabajados priorizando este tipo de recurso.

Resultados de la estrategia con microcontenidos

Prueba estadística: Esta investigación es de índole cualitativo, sin embargo, se decidió aplicar una prueba estadística para medir el tamaño del efecto producido por la intervención. Para esto, se aplicó la prueba estadística de Cohen's d^5 a los valores obtenidos en la prueba final, en contraste con el diagnóstico aplicado. El tamaño del efecto estimado fue de $d = 0.27$, lo que corresponde a un efecto pequeño. Esto sugiere que, aunque la mejora en notas existe, su magnitud es reducida y posiblemente impactada en parte por el tamaño de la muestra.

A los estudiantes se les realizó un diagnóstico inicial, una prueba corta al finalizar cada uno de los cuatro módulos y una prueba final. En la Tabla 2. Se muestran los resultados de varias métricas como las notas promedio y mediana para cada módulo, el diagnóstico y comprobación final.

⁵ Cohen's d es una prueba estadística que corresponde a una medida de efecto. Se usa para comparar la diferencia entre dos grupos. En este artículo se aplica para comparar el grupo antes y después de llevar los microcontenidos.

Tabla 2. Resultados de la experiencia didáctica con microcontenidos de aprendizaje agrupados por módulos temático (Elaboración propia)

Métrica	Módulo 1- Algoritmos	Módulo 2- Variables	Módulo 3- Instancias y Objetos	Módulo 4- Funciones	Diagnóstico	Comprobación Final	Mejora registrada en prueba final vs diagnóstico
Muestra completa							
Promedio	93.91	82.13	92.39	88.70	67.63	73.54	+5.90
Mediana	100	100	100	100	66.67	77.36	+10.69
Estudiantes CTP							
Promedio	91.67	67.59	89.58	78.33	57.41	64.53	+7.12
Mediana	100	77.78	100	100	55.56	69.44	+13.88
Estudiantes WGJ							
Promedio	96.36	97.98	95.45	100	78.79	83.36	+4.57
Mediana	100	100	100	100	88.89	90.57	+1.68

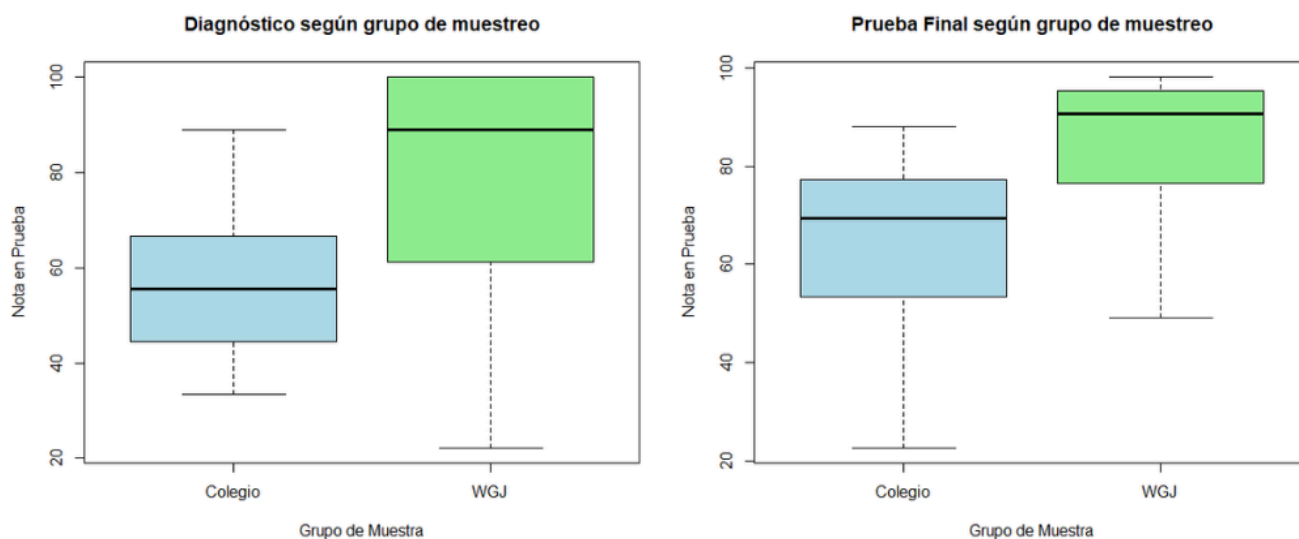
La Tabla 2, refleja el promedio y mediana de las notas de los estudiantes para cada una de las pruebas aplicadas por módulo de microcontenido, diagnóstico inicial y prueba final. En la última columna se obtiene la diferencia en puntos de la prueba final según el diagnóstico inicial. Registrando aumentos puntuales en la comparativa de conocimiento realizada al final del experimento respecto a la prueba de entrada.

Al comparar los valores de la Tabla 2, se observa cómo posterior a la estrategia con los microcontenidos, se obtuvo una mejoría de **5.90** puntos en las notas de los estudiantes: **7.12** para los estudiantes de colegio y **4.57** para las estudiantes de la WGJ. De igual manera, se mejoró la mediana de nota en todos los grupos, registrando una mejora de **13.88** puntos en los estudiantes de colegio, **1.68** puntos en las estudiantes de la WGJ y de **10.69** puntos para la muestra completa. Estos dos resultados muestran que en general la base de estudiantes mejoró las notas posterior a la intervención.

En las gráficas de la Figura 7 y de la Tabla 2, se observa cómo en general las estudiantes de la WGJ entran a la experiencia con un mejor rendimiento. Sin embargo,

después de llevar los microcontenidos, ambos grupos registran mejoras en la prueba de conceptos.

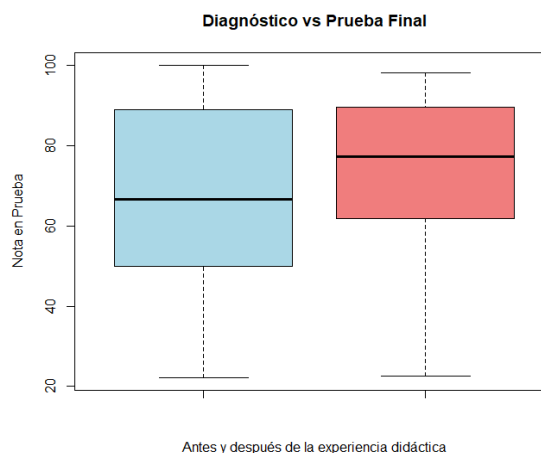
Figura 7. Resultados de grupos al aplicar prueba diagnóstica.



Nota. Gráfico desarrollado con el comando `bbox`, paquete `ggplot2` del lenguaje de programación R. Elaboración propia

En la Figura 8, se muestra cómo más allá de la comparación estadística de medianas, los resultados evidencian una mejora general en los puntajes mínimos. Es decir, en el diagnóstico inicial se observaron calificaciones más bajas, mientras que en la prueba final esas brechas se redujeron y ningún estudiante obtuvo resultados tan desfavorables en la prueba. Así, los microcontenidos contribuyeron especialmente a reforzar los aprendizajes de quienes presentaban mayores dificultades al inicio, elevando el piso de desempeño del grupo y promoviendo una mayor homogeneidad en los resultados.

Figura 8. Resultados antes y después de intervención



Nota: Gráfico desarrollado con paquete `ggplot2`, comando `bbox` del lenguaje de programación R. Elaboración propia

Resultados de la Observación

Adicionalmente, se solicitó a los participantes resolver un problema de programación simple tomado del juez en línea *BeeCrowd*⁶, el cuál posee un repositorio grande de recursos de programación. Los problemas que podían seleccionar para resolver fueron los siguientes (Tabla 3):

Tabla 3. Problemas de algoritmia desarrollados por los estudiantes en la evaluación

Enunciado Ejercicio BeeCrowd	Ejemplo de Respuesta	Observación: El estudiante es capaz de:
<p>ÁREA DE UN CÍRCULO - (Código BeeCrowd 1002)</p> <p>El programa recibe el radio de un círculo y calcula e imprime el área. La fórmula para calcular el área de una circunferencia está definida como $A = \pi \cdot R^2$. Para este problema considera que $\pi = 3.14159$</p>	<pre>Definir PI como 3.14159 Escribir "Ingrese el radio del círculo:" Leer R Área ← PI * R * R Escribir "El área del círculo es: ", Área Fin</pre>	<ul style="list-style-type: none"> -Declarar de variables. -Operaciones de entrada y salida de datos -Capaz de trasladar una fórmula matemática a pseudocódigo -Secuencia lógica de pasos.
<p>PROMEDIO (Código BeeCrowd 1005)</p> <p>Lea dos valores correspondientes a las notas de un estudiante. Luego, calcula el promedio del estudiante, considerando que la nota A tiene peso 3.5 y B tiene peso 7.5. Cada nota puede ir de cero a diez.</p>	<pre>función PromedioPonderado Leer A Leer B promedio = (A * 3.5 + B * 7.5) / 11 Escribir promedio con 5 decimales Fin Algoritmo</pre>	<ul style="list-style-type: none"> -Definición de funciones simples -Operaciones de entrada de datos -Estructura algorítmica clara. -Precisión y formato de salida.
<p>SALARIO CON BONIFICACIÓN (Código BeeCrowd 1007)</p> <p>Haz un programa que lea el nombre de un vendedor, su salario y el total de ventas realizadas en el mes. Este vendedor recibe un 15% de bonificación sobre todas las ventas realizadas. Escribe el salario total al final del mes.</p>	<pre>nombreVendedor = String; salarioVendedor = Int; totalVentasRealizadas = Int; bonificacionVendedor = Int; funcion calcularSalario (nombreVendedor, salarioVendedor, totalVentasRealizadas) { bonificacionVendedor = totalVentasRealizadas / 85; bonificacionVendedor += salarioVendedor; return salarioVendedor }</pre>	<ul style="list-style-type: none"> Declaración de variables con tipo (aunque la sintaxis no sea exactamente correcta) Definición de función con parámetros y valores de retorno. Asignación de valores.
<p>EL MAYOR - (Código BeeCrowd 1013)</p> <p>Haz un programa que lea 3 valores enteros e imprima el mayor de ellos,</p>	<pre>Escribir "Escriba el primer número" Leer A Escribir "Escriba el segundo número" Leer B Escribir "Escriba el tercer número" Leer C if (A>=B) and (A>=C) entonces A es el mayor else if (B>=A) and (B>=C) entonces B es el mayor else C es el mayor End</pre>	<ul style="list-style-type: none"> -Entradas y salidas de datos. -Condicionales -Lógica en conectores -Algoritmia básica.

⁶ Los problemas de programación seleccionados (Códigos: 1002, 1005, 1007, 1013) se encuentran catalogados como dificultad baja o de entrada en la plataforma: <https://judge.beecrowd.com>.

De los 24 estudiantes que completaron el curso, 16 contestaron alguna de las preguntas de algoritmia de la Tabla 3; 2 personas optaron por resolver el problema *Área del Círculo*, 4 seleccionaron el problema del *Promedio*, 2 personas seleccionaron el problema de *Cálculo de bonificación* y 8 personas completaron el ejercicio: *Mayor de tres*. Las otras 8 personas restantes optaron por dejar en blanco la pregunta.

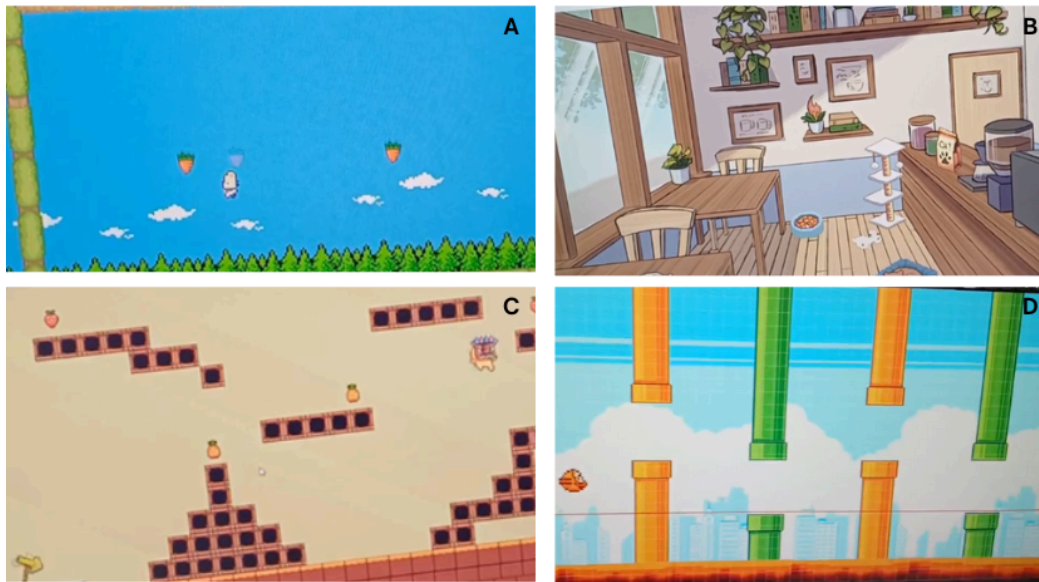
En la Tabla 3, se incluye un ejemplo de respuesta obtenida por pregunta. Cabe destacar que al tratarse de pseudocódigo, las respuestas varían, lo que se manifiesta en que algunas se relacionan más con un lenguaje de programación (incluyendo tipos de datos por ejemplo), mientras otras se acercan más a lenguaje natural o a lenguaje matemático. No obstante, la algoritmia para la resolución de problemas es clara, resuelve el problema y tiene estructura, que es lo que se buscaba con este ejercicio. Las 16 personas ($\frac{2}{3}$) que contestaron esta pregunta abierta lograron esbozar de forma correcta una solución válida a un problema de programación básico. Considerando nuevamente que en la prueba diagnóstica el 100% de los estudiantes declaró no conocer lenguajes de programación.

Finalmente, además de los resultados de las comprobaciones a los estudiantes, se les brindó espacio para desarrollar un proyecto propio en Game Maker en un espacio de laboratorio de 3 horas con los conocimientos aprendidos. Se trabajó en grupos de 2 o 3 personas. Algunos de los resultados se muestran a continuación (Figura 9).

En las 3 horas de tiempo, ninguno de los juegos se terminó, pero se avanzaron prototipos interesantes y creativos no solo desde la parte estética sino desde la innovación. Por ejemplo, conceptos como el manejo de mouse no fueron enseñados en los módulos de microcontenidos, pero los estudiantes en taller experimentaron con esa funcionalidad. La mayoría de los grupos lograron implementar personajes que se movieran en pantalla e interactuar con objetos a nivel básico. Esto es importante porque lograron llevar conceptos del curso a un producto de interés. Las ideas y objetivos de juego que plantean son diferentes entre sí. Recordando conceptos como: algoritmo, instrucción, variables y condicionales. Aunado a esto, dado el tipo de

motor empleado, aplicaron también conceptos como objetos, instancias y eventos/métodos.

Figura 10. Ejemplos de juegos desarrollados por los estudiantes con Game Maker



Nota: a) Juego de conejo que atrapa zanahorias. b) Juego de reposicionamiento de objetos a través del click y arrastre c) Juego de recolección de piñas al rebotar en las paredes y d) Juego estilo flappy-bird
Fotografías de autoría propia tomadas a los proyectos estudiantiles desarrollados en las sesiones prácticas en laboratorio.

Discusión

Las intervenciones didácticas que usan videojuegos para la enseñanza de la programación son bastante utilizadas ya que favorecen la experimentación y aprendizaje situado. En este proyecto el enfoque estuvo centrado no solo en la instrumentalización del uso de motores de programación de videojuegos, sino también, en la comprensión conceptual de los principios de programación que subyacen a las funcionalidades aplicadas. Este énfasis resultó importante porque respondió a lo planteado por Papert (1984) con su teoría del construccionismo, donde los aprendizajes se potencian cuando las personas crean objetos que les son significativos y a partir de eso llegan a comprender los procesos detrás de su construcción.

La estrategia de microcontenidos buscó precisamente favorecer esa conceptualización progresiva, permitiendo a los y las estudiantes asimilar los fundamentos en pequeñas unidades cognitivamente manejables. Desde la perspectiva del *microlearning*, la fragmentación facilitó la apropiación del conocimiento,

especialmente para personas sin formación previa en área computacional. Además, al enfocarse en conceptos se garantiza un aprendizaje más trascendente que posteriormente puede ser utilizado con otros entornos o lenguajes, rescatando así la teoría planteada por Wing (2006).

En general, para la muestra de trabajo, los resultados evidenciaron una mejora general respecto al diagnóstico inicial, con una reducción de la brecha entre los niveles bajos y altos. Además, otro de los logros principales recae en la resolución de problemas de algoritmia y extrapolación de conceptos —con dos tercios de los estudiantes animándose a resolver problemas de esta índole—, lo cual denota un entendimiento de varios conceptos importantes de los módulos para la programación.

El espacio de creación de videojuegos, durante la jornada presencial, funcionó como un entorno de aprendizaje creativo. No solo se observó apropiación de conceptos, sino también una buena actitud respecto al tema. Al tratarse de un tema interesante que los estudiantes pueden trabajar entre pares, las personas estudiantes de la muestra superaron en gran medida esa matemafobia que Papert (1984) menciona hacia la herramienta y también hacia la materia. En este marco, aunque se enfrentaron a obstáculos y frustraciones, la disposición percibida en general fue mucho mejor que antes de la aplicación de la estrategia.

Un aspecto importante que cabe resaltar es que aunque la mayoría de los estudiantes dice preferir los recursos en video para aprender, en la práctica se observó que las guías escritas con imágenes resultan de mucha utilidad. Por ejemplo, en el laboratorio, cuando los estudiantes necesitaban revisar o repasar algún contenido, buscaban la guía escrita antes que los videos. Por lo tanto, una combinación de métodos es ideal.

En el caso de las participantes de la Women Game Jam (WGJ), los resultados reflejaron el papel determinante de la motivación intrínseca. Su mejor desempeño puede vincularse con una doble motivación: aprender y prepararse para la maratónica. La posterior participación de varias de ellas en una segunda Game Jam (Game Jam Plus), de carácter más competitivo, evidenció la consolidación de un proceso de empoderamiento técnico y creativo, altamente valioso y coherente con los objetivos de vinculación que promueve la CIDeV.

Así, la estrategia con microcontenidos resultó bastante eficaz para transmitir los conceptos, lo cual quedó evidenciado no solo en las notas de los estudiantes en las pruebas comprobatorias, sino en la actitud demostrada a la materia y la anuencia a extrapolar lo aprendido a otros problemas de programación como los del juez en línea: *Beecrowd*. En este sentido, el aprender programación haciendo videojuegos, funcionó bastante bien para reducir el miedo o repulsión a la programación que tienen algunos estudiantes que participan de actividades de la CIDEV, reforzando la teoría del abordaje y uso de estos en ambientes educativos. En este caso con personas de arte, música y estudiantes de colegio.

Conclusiones

La estrategia de microcontenidos desarrollada para el aprendizaje de la programación con videojuegos propició que la experiencia de las personas estudiantes fuera entretenida y creativa. Además, las personas estudiantes mostraron buena disposición y anuencia a aprender programación y codificar, superando las barreras y resistencia usuales a la materia. Asimismo, la experiencia se enriquece aún más con la presencialidad, cuando las personas trabajan en equipos, pues se ayudan y se motivan entre pares.

Los microcontenidos en video resultaron muy útiles para transmitir conceptos teóricos, lo cual se vió reflejado en las notas de los estudiantes. Sin embargo, para procesos guiados y prácticos en el uso del software resultan más útiles las guías de texto e imágenes. Eso se refleja en el hecho de que en las sesiones presenciales se observó que cuando los estudiantes requerían información de comandos o uso en general de Game Maker buscaban específicamente el material escrito, por su facilidad para llegar al punto de interés con rapidez.

Algunas de las limitaciones presentadas se dieron por la falta de competencias en el uso del hardware o software (por ejemplo: manejo de mouse o navegación por carpetas) por parte de un grupo pequeño de estudiantes. Al trabajar en equipos, los estudiantes que tenían estos conocimientos ayudaron a los que no y de esta forma, se lograron completar los ejercicios. Otra limitante que se tuvo es que no todos los estudiantes que iniciaron con los módulos finalizaron los mismos y por lo tanto, no se

podieron incluir en la muestra final aunque sí asistieron a las sesiones presenciales y realizaron aportes valiosos a sus respectivos equipos de trabajo.

Como trabajo futuro, se quiere aplicar la misma estrategia de microcontenidos con una muestra diferente de estudiantes, para determinar si el efecto es trasladable a otros grupos fuera de la CIDeV. Esta misma estrategia, se quiere probar con adultos universitarios de otras carreras, no necesariamente relacionados con el tema de videojuegos, para analizar cómo se desempeña el videojuego como motivador en estos grupos.

Referencias Bibliográficas

- Acosta, M.; Cabrera, R.; Montaguano, J.; León, W.; Morocho, J. y Toala A. (2020) GameMaker as an instrument to create educational resources based on game elements. EDULEARN20 Proceedings, pp. 4185-4192.
- Alamina, M.A.F; Kavishcan, V.; Jayaratne, W.M.P.B.B.; Kannangara, K.K.D.S.N; Aamil, A.A.; Adikari, Achini. (2025). A Self-Paced Generative AI Chatbot Approach for Programming Skills Training. ArXiv. <https://arxiv.org/html/2503.16486v1#bib.bib4>
- Angeli, Charoula.; Giannakos, Michail. (2020). Computational thinking education: Issues and challenges, Computers in Human Behavior, Volume 105, ISSN 0747-5632, <https://doi.org/10.1016/j.chb.2019.106185>.
- AYYILDIZ, hamdi. (2020). Game Development Software Tools in Higher Educational Institutions: Experience of Ukraine, Turkey and Bulgaria. Information Technologies and Learning Tools. <https://doi.org/10.33407/ITLT.V78I4.3370>
- Capecchi, Irene, Tommaso Borghini, Elena Barbierato, Andrea Guazzini, Elena Serritella, Tommaso Raimondi, Claudio Saragosa, y Iacopo Bernetti. (2022). The Combination of Serious Gaming and Immersive Virtual Reality through the Constructivist Approach: An Application to Teaching Architecture. Education Sciences 12, no. 8: 536. <https://doi.org/10.3390/educsci12080536>
- Céspedes-Vindas, A. y Mora-Zamora, R. (2024). Counterspaces for Women in Costa Rica: Case of study of CIDeV Game Jams. 22nd LACCEI International Multi-Conference for Engineering, Education, and Technology San Jose – COSTA RICA, July 17 - 19, 2024.
- CID - Centro de Investigación y Desarrollo (2024). Compilación de Investigaciones en Ciencias de la Educación “Enseñanza y Aprendizaje”. ISBN: 978-99989-64-60-0. DOI: https://doi.org/10.37811/cli_w1012

- Delgado, P., Anmarkrud, Ø., Avila, V. (2022). Learning from text and video blogs: comprehension effects on secondary school students. *Educ Inf Technol* 27, 5249–5275 (2022). <https://doi.org/10.1007/s10639-021-10819-2>
- Fidan, Mustafa (2022). The effects of microlearning-supported flipped classrooms on pre-service teacher's learning performance, motivation and engagement. *Education and Information Technologies*. Vol 28. Pp 12687-12714
- Gascho Rempel, Hannah., Slebodnik, Maribeth. (2024). *Creating Online Tutorials: A Practical Guide for Librarians*. Rowman & Littlefield. London.
- Gajewski, Sebastian; El Mawas, Nour; Heutte, Jean. (2022). A systematic literature review of game design tools. *Proceedings of the 14th International Conference on Computer Supported Education, 2022, A distance, France*. pp.404-414, [10.5220/0011137800003182](https://doi.org/10.5220/0011137800003182)
- Givi Efgivia, M., Adora Rinanda, R.Y., Surinayi., Hidayat, Ang., Maulana, Irfan., Budiarjo, Anthon. (2021). Analysis of Constructivism Learning Theory. *Proceedings of the 1st UMGESHIC International Seminar on Health, Social Science and Humanities (UMGESHIC-ISHSH 2020)*. Atlantis Press. Vol. 585
- Holenko Dlab, M., Hoic-Bozic, N. (2021). Effectiveness of game development-based learning for acquiring programming skills in lower secondary education in Croatia. *Educ Inf Technol* 26, 4433–4456. <https://doi.org/10.1007/s10639-021-10471-w>
- Lewis, N., Lewis, R. & Luca, C. (2023). Impact of video-based learning in business statistics: a longitudinal study. *Humanit Soc Sci Commun* 10, 146. <https://doi.org/10.1057/s41599-023-01634-w>
- Lodi, M., Martini, S. (2021). Computational Thinking, Between Papert and Wing. *Sci & Educ* 30, 883–908. <https://doi.org/10.1007/s11191-021-00202-5>
- Matthes, Eric. (2023). *Python Crash Course, 3rd Edition: A Hands-On, Project-Based Introduction to Programming*. No Starch Press. San Francisco, California.
- Mora-Zamora, R. y Céspedes-Vindas (2024). Game Jams as a STEAM learning experience to complement computer engineering curricula. *CONCAPAN XLII*. San José, Costa Rica. Nov 27-29, 2024
- Nieves Loja, Gerardo Miguel., Pérez, Vicente Eduardo Benítez. (2021). Paradigmas emergentes. *Diálogos con Edgar Morin en un horizonte intercultural*. *Aufklärung: revista de filosofía*. Vol. 8, No. 3. Pp 21-32
- Olmedo, Noelia, Farrerons, Oscar, Pujol, Ana. (2021). *Constructivist learning models in training programs*. *Omnia science*. Universidad Politécnica de Cataluña, España. ISBN: 978-84-122028-9-2.

- Osman, Erol (2020). How do Students' Attitudes Towards Programming and Self-Efficacy in Programming.
- Papert, S. (1984). *Desafío de la Mente*. Ediciones Galápago
- Reynaldo, Charles., Christian, Ryan., Hosea, Hansel., Gunawan, Alexander. (2021). Using Video Games to Improve Capabilities in Decision Making and Cognitive Skill: A Literature Review. *Procedia Computer Science*. Volume 179. Pp 211-221, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2020.12.027>.
- Rodríguez, Erika., Trujillo, José Saúl., Vargas, Diana., Corredor, Alba., Gallego, Lina María. (2018). El paradigma emergente y la educación. *Educación*. Vol. 29 (N° 10)
- Rodríguez, E. R. (2020). Tutoriales de Youtube como estrategia de aprendizaje no formal en estudiantes universitarios. *RIDE Revista Iberoamericana Para La Investigación Y El Desarrollo Educativo*, 11(21). <https://doi.org/10.23913/ride.v11i21.797>
- Rugelj, J., & Lapina, M. (2019, May). Game design-based learning of programming. In *Proceedings of SLET-2019–International Scientific Conference Innovative Approaches to the Application of Digital Technologies in Education and Research* (pp. 20-23).
- Sandí, J. C. y Sanz, C. V. (2018). Análisis comparativo de juegos serios educativos. Indagación sobre sus posibilidades para la adquisición de competencias tecnológicas en la formación del profesorado. (Tesis de posgrado). Universidad Nacional de La Plata (UNLP), Buenos Aires, Argentina. Recuperado de <http://hdl.handle.net/10915/65653>
- Shakib, Sadat Shanto; Zishan, Ahmed; y Akinul, Islam Jony. (2025). Generative AI for Programming Education: Can ChatGPT Facilitate the Acquisition of Fundamental Programming Skills for Novices? In *Proceedings of the 3rd International Conference on Computing Advancements (ICCA '24)*. Association for Computing Machinery, New York, NY, USA, 685–692. <https://doi.org/10.1145/3723178.3723268>
- Thillainadesan, Janani, Le Couteur, David G., Haq, Inam, Wilkinson, Tim J. (2022). When I say... Microlearning. *Med Educ*. 2022. Vol 56. pp 791-792
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Zamata-Aguirre, H. R., Choquehuanca-Quispe, W., Machaca-Huamanhorcco, E., Salas Begazo, A. N., & Bernedo Málaga, V. W. (2023). Towards the development of learning through microlearning. *Ciencia Latina Revista Científica Multidisciplinar*, 7(1), 3939-3954.