

Dirección de Orientación Vocacional DOV-Exactas, UBA. Docente de la materia Orientación Vocacional Ocupacional de la Carrera de Psicología de la Universidad de Palermo. Docente y Coordinadora técnica de la Especialización en Orientación Vocacional y Educativa (APORA – UNTREF). Atención Clínica y Orientación Vocacional en consultorio. • **María Sara Müller.** Doctoranda del Programa Interuniversitario de Doctorado en Educación (UNTREF, UNSAM, UNLA). Diplomatura en Edu-

cación Permanente de Jóvenes y Adultos (UMET). Profesora en Docencia Superior (UTN). Magíster y Especialista en Educación, lenguajes y medios (UNSAM). Licenciada en Comunicación Audiovisual (UNSAM-2001). Productora y Directora de radio y televisión (ISER-1996). Se desempeña como docente en UP, UB y CENS 69. Directora del CENS 26 -Secundario para jóvenes y adultos- Ministerio de Educación- GCBA.

PSeint como herramienta mediadora en la creación de juegos matemáticos

Fecha de recepción: julio 2021
Fecha de aceptación: septiembre 2021
Versión final: noviembre 2021

Graciela Djeboglian, Paula Ithurralde y María Gabriela Galli (*)

Resumen: Abordar problemas interdisciplinarios desde algunas asignaturas de la tecnicatura en Informática Aplicada (UTN-INSPT), pone en evidencia la necesidad de rediseñar los procesos de enseñanza con el propósito de promover en los estudiantes aprendizajes significativos. Específicamente, en este trabajo se relata una experiencia donde los estudiantes crearon juegos digitales matemáticos, desarrollando y fortaleciendo su pensamiento algorítmico con el software PSeint en la asignatura Laboratorio. La propuesta muestra cómo la incorporación de distintas tecnologías en el aula, bajo una modalidad constructivista, cooperativa y colaborativa, genera clases más dinámicas y motivan a los estudiantes, aprovechando los beneficios que ofrece la programación en los procesos de aprendizaje de las diferentes áreas disciplinares.

Palabras clave: Aprendizaje colaborativo - didáctica - digital - juego - matemática - programación.

[Resúmenes en inglés y portugués en la página 167]

Introducción

La Tecnicatura en Informática Aplicada, del Instituto Nacional Superior del Profesorado Técnico – Universidad Tecnológica Nacional (UTN-INSPT, Ciudad Autónoma de Buenos Aires) forma estudiantes en el desarrollo, mantenimiento y programación de sistemas informáticos. En su diseño curricular, dentro de las asignaturas específicas de primer año, se encuentran Programación I y Laboratorio y, destacamos dentro de las troncales a Análisis Matemático I y Álgebra, Probabilidad y Estadística. Estas comparten como objetivos el desarrollo del pensamiento crítico, lógico y algorítmico; la comprensión de enunciados, el modelaje de situaciones; el análisis, la interpretación y el procesamiento de datos y la resolución de problemas.

La programación es el eje conductor de la tecnicatura. Al inicio de la cursada, desde la asignatura Laboratorio, se introducen contenidos básicos de programación entre los cuales se incluyen el uso de algoritmos y pseudocódigo con el *software* PSeint (abreviatura de *Pseudocode Interpreter*), con el propósito de que los estudiantes adquieran las estructuras lógicas de programación. Por otro lado, tanto en Análisis Matemático I como en Álgebra, Probabilidad y Estadística, en sus primeras unidades introductorias, se retoman conceptos del nivel secundario

los que requieren la puesta en marcha de procedimientos y/o resolución de situaciones problemáticas para construir sobre ellos conocimientos más complejos.

Como equipo docente, nuestra experiencia revela que los estudiantes durante los primeros meses del cursado de las asignaturas presentan dificultades en comprender enunciados e identificar las diferencias entre qué solicita y cómo se resuelve una situación problemática; en buscar patrones, estimar y validar datos, en identificar errores, modelar situaciones, entre otras. Particularmente, en programación se observa que muchos de ellos poseen escasas habilidades en la disciplina ya que, programar, no solo requiere manejar conceptos acerca qué es una variable o una constante, sino también desarrollar una serie de habilidades que involucren estructuras para el desarrollo de algoritmos más complejos. También, se les presenta otra dificultad al tener que enfrentarse a entornos de desarrollo integrado y a la sintaxis del propio lenguaje. Todas estas tareas combinadas conllevan una carga emocional-cognitiva que muchas veces termina en el fracaso o fin de la cursada.

Frente a estas dificultades, y ante todo a la falta o escasa experiencia previa que tienen los estudiantes al programar, desde la asignatura Laboratorio, se ha propuesto la enseñanza de la programación a través de pseudo-

código, como base y complemento de las estructuras más complejas que aprenden en Programación I. La propuesta consistía en que los estudiantes desarrollaran un proyecto de programación libre, donde el 90% optó por juegos matemáticos. De esta manera, se ha logrado que aquellos rescaten la funcionalidad del juego como herramienta social y cognitiva, poniendo en juego habilidades y competencias matemáticas y específicas de programación que favorezcan el anclaje de nuevas estructuras a partir de sus contenidos previos. Es decir, se ha fomentado el *aprender haciendo* y *desarrollando productos* que se constituirán en insumo para el aprendizaje de los mismos estudiantes y de terceros.

Marco teórico

Programar interpela al sujeto-programador a revisar constantemente el desarrollo de su código y a aprender del error para que el *software* cumpla con su objetivo, independientemente del lenguaje que se utilice. Por otro lado, el juego en general interpela al usuario en la toma de decisiones a partir de los acontecimientos que se le presentan, donde el error se constituye en fuente de aprendizaje para llegar a la meta. Para ganar un juego, generalmente se ponen en práctica habilidades tales como contabilizar, deducir, descubrir, planificar, observar, probar, generalizar, las que tienen mucho que ver con la matemática. Es por ello que, en esta última disciplina, el juego permite comprender y explicar fenómenos, reflexionar, tomar decisiones, analizar las soluciones óptimas, validar procedimientos, aspectos que, a su vez, son compartidos por la programación.

Con la incorporación en las prácticas pedagógicas del juego en general, o de los juegos digitales en particular, como fenómeno sociocultural masivo, se pueden crear situaciones con alto valor educativo donde los estudiantes pueden fortalecer y adquirir habilidades cognitivas, sociales, emocionales, de toma de decisiones, pensamiento estratégico, análisis interpretativo, gestión de recursos, aumento de la creatividad y motivación, entre otras (Delgado, 2014; Li, Chen y Chen, 2016; Revuelta y Antequera, 2011; Westerveld, 2017). Profundizando en el desarrollo de juegos digitales, centramos la mirada en el programador como responsable de escribir el código para que el juego se ejecute según el objetivo fijado. Esto requiere de un trabajo en equipo en sus instancias de planificación, diseño, elección de *software*, establecimiento de las mecánicas, dinámicas y estética, escritura del código y testeado en búsqueda de errores y mejoras en la jugabilidad.

Basamos nuestra propuesta en Piaget (1956/67) a partir del aporte del juego en la construcción de una amplia red de dispositivos que permiten al sujeto la asimilación de la realidad, incorporándola para revivirla, dominarla y comprenderla. En la teoría sociocultural de Vigotsky (1934) a raíz de que el sujeto construye significados a partir de las contribuciones de los objetos culturales e interactuando con otras personas. También, en el aprendizaje por descubrimiento de Bruner (1960), promoviendo que el sujeto construya nuevos conocimientos por sí mismo a partir de sus ideas preexistentes; en el aprendizaje significativo de Ausubel, donde los nuevos

significados elaborados por el sujeto son el producto de la relación entre el material o contenido nuevo potencialmente significativo y su predisposición subjetiva, tanto emocional como cognitiva, que le permitirán asociar la información nueva con sus esquemas preexistentes. En esta línea constructivista, destacamos los aportes de la cognición situada, es decir, que el conocimiento es parte y producto del contexto y la cultura en que se desarrolla y utiliza (Brown y Cole, 2001) y de la cognición distribuida (Salomon, 1993), donde los artefactos o agentes del entorno contribuyen a la construcción de conocimiento afectando la concepción de qué, cómo y por qué se necesita conocer (Hutchins, 1995).

Desde una perspectiva constructivista en general, donde el aprendizaje es entendido como un proceso en el cual el sujeto construye activamente nuevos conceptos sobre ideas preexistentes, podemos decir que aprender a programar se enmarca en esta línea con un sujeto protagonista del proceso, donde continuamente su conocimiento se reelabora y amplía.

PSeint y sus aportes al proceso de enseñanza

En el mundo de la programación existen distintos paradigmas como manera de pensar un problema y su solución: i) el paradigma de programación estructurada, en el cual se utilizan únicamente subrutinas y tres estructuras básicas de control, secuencias, iteración y selección; ii) el orientado a objetos, donde cada objeto ofrece una funcionalidad especial, tiene un determinado estado, comportamiento e identidad y, iii) el funcional, basado en la idea de función, que podríamos decir es análoga al concepto de funciones matemáticas, en la que se usan principalmente funciones puras y valores inmutables. El paradigma estructurado es al cual pertenece la herramienta PSeint como introducción a la programación.

PSeint es intérprete de pseudocódigo (o falso código: método usado para expresar de forma ordenada los distintos pasos o procesos de un programa que dan solución a un problema específico (algoritmo), con el propósito de ser interpretado a simple vista por el usuario) que les permite a los usuarios de habla hispana, con o sin experiencia en este campo, escribir y ejecutar algoritmos usando pseudocódigo y/o diagramas, tanto de flujo clásico como de Nassi-Shneiderman. Su propósito es contribuir al desarrollo del pensamiento lógico y algorítmico de sus usuarios.

Las características más significativas de esta interfaz son, por ejemplo, la posibilidad de insertar texto y sintaxis mediante autocompletado, resaltado de bloques lógicos y coloreado de sintaxis. Asimismo, ofrece plantillas de comandos, indentado inteligente, listados de operadores, funciones, tipo de datos, entre otras. Es multiplataforma (Microsoft Windows, GNU/Linux y Mac OS X), es libre y gratuito (Licencia GPLv2) y permite convertir el código a numerosos lenguajes de programación, C, C++, C#, Java, JavaScript, Matlab, Pascal, PHP, Python 2, Python 3, QBasic, Visual Basic, con una codificación bastante aproximada.

Entre sus características resaltamos que señala los errores de sintaxis tanto en tiempo real como en el de ejecu-

ción y ofrece descripciones detalladas de cada uno de ellos con sus causas y soluciones posibles. Interpreta los algoritmos desarrollados, permitiendo modificarlos sin reingresar datos. Habilita el cambio de uno o más datos posteriores a la ejecución del programa para el análisis de variación de resultados y también, permite deshacer una ejecución para reiniciarla o repetirla desde un punto arbitrario. Además, tiene como potencial la opción de ejecutar el algoritmo paso a paso, permitiendo controlar variables y expresiones, con la posibilidad de confeccionar automáticamente una prueba de escritorio detallando las acciones realizadas. Es decir, le facilita al usuario tanto la escritura como la comprensión de este a partir de la detección de fallos.

Está diseñado para que aprendan conceptos tales como: variable, constante y tipos de datos utilizando estructuras selectivas, repetitivas, arreglos, funciones y procedimientos. También, al ser en español, subsana uno de los problemas a los que se enfrentan los estudiantes al encontrarse con palabras reservadas generalmente en inglés, brindando soluciones más comprensibles a través de estructuras que se relacionan más con su lenguaje cotidiano.

Con el propósito de acercar a los estudiantes a las estructuras de programación e incrementar sus conocimientos lógicos, a continuación, se detalla la intervención implementada.

Relato de la experiencia

Esta experiencia sentó sus bases en los espacios curriculares vinculados con matemática, donde los estudiantes trabajaron con juegos analógicos de estrategia de tipo numérico, con el propósito de revisar y fortalecer conceptos vistos en su formación secundaria, entre ellos se mencionan conjuntos numéricos, ecuaciones, criterios de divisibilidad, operaciones básicas. Estos desafíos enmarcados en propuestas lúdicas, contribuyeron al desarrollo del pensamiento lógico y deductivo, se socializaron propiedades y generalizaciones y, ante todo, rompieron con los cánones tradicionales de una clase de nivel superior, despertando la motivación y el interés, fortaleciendo el intercambio estudiante-estudiante y estudiante-docente en marcos de trabajo colaborativo. Poco tiempo después, en la materia Laboratorio se les propuso a los estudiantes el desafío de elaborar un proyecto de programación libre en PSeint. Como se expuso anteriormente, 90% eligió desarrollar juegos y a partir de la experiencia con matemática, los hicieron aplicados a esa disciplina. Algunos optaron por replicar los mismos que habían experimentado, otros utilizaron la misma mecánica recreando la situación, mientras que algunos aportaron propuestas nuevas.

En el interior de cada grupo, durante quince días, los estudiantes analizaron la lógica de cada juego experimentado en matemática y diseñaron, en papel, un bosquejo del que iban a programar, identificando los tipos de datos de entrada y salida y su forma de procesamiento, las estructuras que se requerían, todos los casos de posibles soluciones y los errores que podrían presentarse. De esta forma, acotaron el problema y diseñaron el plan de acción.

A partir de la interacción, de forma espontánea, cada estudiante adquirió un rol distinto en función de sus habilidades: organizador general, programador, analista y testeador, sin embargo, todas las etapas del proyecto fueron nutridas por todos sus integrantes.

Al momento de escribir el pseudocódigo del juego en PSeint, sobre la base del algoritmo diseñado en papel, los estudiantes se enfrentaron con estructuras conocidas y otras a descubrir, lo que implicó investigar en el tema para subsanar los emergentes. Una vez finalizado el pseudocódigo se procedió a su ejecución para analizar su funcionamiento correcto y allí los estudiantes quisieron resumir sus líneas para obtener un programa eficiente, con lo cual trabajaron con nuevas funciones por sobre las básicas brindadas. Estos descubrimientos fueron compartidos con los integrantes de otros grupos, debatidos e incorporados a las producciones en función de los desarrollos, evidenciándose un gran trabajo colaborativo. También, vale resaltar, que todos los juegos fueron testeados por todos los grupos, aportando mejoras en todos los desarrollos.

Este trabajo por proyectos permitió integrar y debatir conceptos de diferentes disciplinas. La ventaja con otros proyectos es que la programación en la misma práctica marca los errores y los aciertos a la hora de hacer funcionar un programa sin la supervisión directa del docente. Este *feedback* inmediato que posibilita la computadora favoreció no solo la exploración y el descubrimiento de los contenidos, sino que les permitió a los estudiantes avanzar a su propio ritmo y generar autonomía y confianza en la aprehensión de los conocimientos.

Conclusiones

En la actualidad, en los niveles de educación obligatoria, la programación comienza a estar presente, ya que programar, colabora con el fortalecimiento del trabajo en equipo, la capacidad de abstracción y de planificación, el razonamiento lógico y algorítmico. Los niños y adolescentes, poco a poco, comienzan a realizar sus primeros desarrollos en entornos tales como Pilas, Alice, Scratch, Gobstones, entre otros, y asocian a la programación con los videojuegos. En esta línea y retomando las palabras de Corsi (2019, p.55) que “no es necesario tener los conocimientos de programación de un desarrollador de *software* profesional para sacarle el provecho a la programación, ya que - incluso creando programas muy básicos para automatizar tareas repetitivas - es posible aumentar la productividad, al reducir los tiempos y comprender mejor los procesos involucrados”, sin distinción de edades, todos podemos sumergirnos en el arte de la programación.

Desde el año 2014, en la Tecnicatura en Informática Aplicada se vienen incorporando juegos digitales en distintos espacios curriculares y con distintas funcionalidades, desde su programación, como aplicación de lenguajes específicos; a la construcción con aplicaciones desarrolladas por terceros o con el uso de juegos comerciales. Esta carrera no forma desarrolladores en videojuegos, pero sí les aporta a sus estudiantes las herramientas necesarias para poder desempeñarse en dis-

tintos sectores de la empresa e industria vinculadas con la informática.

Consideramos que la experiencia descrita, iniciada con la experimentación con juegos matemáticos donde los estudiantes recuperan conceptos previos y más allá de ganar, se tuvo que descubrir su lógica para su posterior programación, les ha permitido vincular y fortalecer conocimientos en distintas áreas, donde el juego, se constituyó en una fuente motivacional y llave para indagar más en la pedagogía lúdica.

El uso de PSeint les ha permitido a los estudiantes comprender mejor qué es un algoritmo, un pseudocódigo y la importancia de trabajar con diagramas de flujo en sus primeros pasos como programadores de sistemas más complejos. Además, ha colaborado con el desarrollo de la lógica, la investigación e incorporación y profundización en el uso de las distintas herramientas de modo autodidacta. Este tipo de propuestas alienta el trabajo colaborativo donde, entre estudiantes y con el docente, se comparten los descubrimientos que emergen como parte del desarrollo del proyecto, activas dinámicas de clase como interés, colaboración e integración de todos los estudiantes. De esta manera, la lógica de programación genera un involucramiento con la tarea por parte de la mayoría de los estudiantes y los motiva en sus desarrollos posteriores.

Referencias

- Brown, K. y Cole, M. (2001). Cultural historical activity theory and the expansion of opportunities for learning after school. En M. J. Packer y M. B. Tappan (Eds.), *Cultural and critical perspectives on human development*. Nueva York: SUNY Press.
- Bruner, J. (1960). Low-Temperature Internal Friction in Face-Centered Cubic and Body-Centered Cubic Metals. *American Physical Society*, 118 (2).
- Corsi, D. (2019). *Estrategias lúdicas para la enseñanza de la programación: un análisis comparativo de su eficacia en la Educación Superior* (Tesis Doctoral). España: UNEX.
- Delgado, J. (2014). *Beneficios de los videojuegos para tratar trastornos infantiles*. Obtenido de www.etapainfantil.com/beneficios-videojuegos-tratar-trastornos-infantiles.
- Hutchins, E. (1995). *Cognition in the wild*. The MIT Press.
- Li, L., Chen, R. y Chen, J. (2016). Playing Action Video Games Improves Visuomotor Control. *Advances in Psychological*, 27(8), 1092-1108.
- Piaget, J. (1956/1967). Programas y métodos de la epistemología genética. En J. Piaget, *Psicología, lógica y comunicación* (págs. 21-118; N. Bastard, Trad.) Buenos Aires: Ediciones Nueva Visión: Colección Ensayos.
- Revuelta, F. y Guerra, J. (15 de Octubre de 2012). ¿Qué aprendo con videojuegos? Una perspectiva de meta-aprendizaje del videojugador. *RED. Revista de Educación a Distancia* (33), 1- 25.
- Salomon, Gavriel (1993), “Introducción” y “No hay distribución sin la cognición de los individuos: un enfoque interactivo dinámico”. En G. Saloman (comp), *Cogniciones distribuidas. Consideraciones psicológicas y educativas*, Buenos Aires: Amorrortu.
- Westerveld, R. (2017). Beneficios y efectos positivos de los videojuegos. *Publicaciones didácticas*.

Abstract: Addressing interdisciplinary problems from some subjects of the Applied Computer Science Technique (UTN-INSPT), highlights the need to redesign teaching processes in order to promote meaningful learning in students. Specifically, in this work an experience is related where the students created mathematical digital games, developing and strengthening their algorithmic thinking with the PSeint software in the Laboratory subject. The proposal shows how the incorporation of different technologies in the classroom, under a constructivist, cooperative and collaborative modality, generates more dynamic classes and motivates students, taking advantage of the benefits that programming offers in the learning processes of the different disciplinary areas.

Keywords: Collaborative learning - didactics - digital - game - mathematics - programming.

Resumo: Neste artigo falamos de um Projeto de Extensão da Universidade Federal de Santa Maria (UFSM) que foi desenvolvido no município de Santa Maria (RS). Apresentamos resumidamente a história do projeto que trata dos truques matemáticos com o objetivo de incentivar o aluno a aprender de forma lúdica. Num segundo momento, apresentamos um exemplo de adaptação de uma manobra em que o aluno atua como um mágico sem ter nenhum conhecimento do processo. É uma atividade chamada quebra-gelo, pois é utilizada no início da apresentação.

Palavras chave: Aprendizagem - aula - ensino - experiência - lúdico - matemática - projeto pedagógico.

(*) **Graciela Djeboglian**. Licenciada en ciencias aplicadas (UTN). Profesora en Disciplinas Industriales, especialidad “Informática Aplicada” y técnica superior en Informática Aplicada (UTN-INSPT). Se desempeña como profesora en UTN-INSPT. • **Paula Ithurralde**. Técnica en Administración y Gestión Universitaria (UBA) Profesora en Disciplinas Industriales, especialidad “Informática Aplicada” y técnica superior en Informática Aplicada (UTN-INSPT). Se desempeña como profesora en UTN-INSPT. • **María Gabriela Galli**. Doctora en Política y Gestión de la Educación Superior (UNTREF). Especialista en Educación y TIC, profesora en Disciplinas Industriales, especialidad “Matemática Aplicada” y técnica Superior en Informática Aplicada (UTN-INSPT). Se desempeña como directora de carrera y profesora en UTN-INSPT, y como coordinadora de área y docente en escuelas del GCBA. Es miembro del equipo de investigación en Tecnología Educativa de UNTREF y de la asociación ALFAS.