

El juego como actividad estructurante en el aprendizaje de la programación

The game as a structuring activity in programming learning

Prof. Alejandro Miños

Consejo de Formación en Educación, Instituto Normal de Enseñanza Técnica, Uruguay
alejandromifa@gmail.com

Prof. Rodrigo Herrera

Dirección General de Educación Secundaria, Uruguay
rodrigoh79@gmail.com

Resumen — Uno de los principales desafíos a los que se enfrenta la institución educativa es que la misma resulte motivante para los alumnos, maximizando el interés en las actividades y las posibilidades de generar aprendizajes en los estudiantes. Una forma de motivar a los alumnos es a través del juego, el cual en el mejor de los casos, se convierte en estructurante de la actividad. Desde esta perspectiva, el juego no es un fin en sí mismo, sino un medio para lograr aprendizajes en los alumnos. Este trabajo describe una secuencia didáctica para el aprendizaje de la programación usando Scratch en enseñanza media básica, con el juego como eje central de la misma, articulando contenidos disciplinares. Se proponen tres juegos, describen los contenidos trabajados, su secuenciación y el tiempo dedicado a cada uno de ellos.

Palabras clave; didáctica de la informática; programación; secuencia didáctica.

Abstract — One of the main challenges that the educational institution faces is that it be motivating for the students, maximizing the interest in the activities and the possibilities of generating learning in the students. One way to motivate students is through games, which in the best of cases, become the structure of the activity. From this perspective, the game is not an end in itself, but a means to achieve learning in students. This work describes a didactic sequence for learning programming using Scratch in basic secondary education, with the game as its central axis, articulating disciplinary contents. Three games are proposed, they describe the contents worked on, their sequencing and the time dedicated to each one of them.

Keywords; informatics didactics; programming; didactic sequence.

I. INTRODUCCIÓN

La informática se ha convertido en una de las grandes áreas de conocimiento a ser estudiada en la enseñanza secundaria, siendo un caso particular el de la enseñanza media básica (tres primeros años de la enseñanza post primaria). En los programas de la asignatura informática una de las unidades temáticas que en general es estudiada, desde la concepción de la informática como disciplina científica [1], lo constituye la programación. La enseñanza de la programación en este nivel en general no es un fin formativo en sí mismo, sino que se constituye en un medio para lograr otros objetivos o competencias. Desde esta perspectiva, en enseñanza media básica no se enseña programación con el fin de lograr formar programadores o personal calificado que luego se inserte en el mercado laboral.

En el caso del sistema educativo uruguayo, la asignatura Taller de Informática está presente en primer y segundo año de enseñanza media básica en la Dirección General de Educación Secundaria, Ciclo básico, con un total de 4 horas semanales de clase (cada hora de clase de 45 minutos) [2]. En este nivel educativo, la inclusión de la programación en el currículo se vincula con el desarrollo de habilidades de pensamiento y lenguaje científico, fomento del pensamiento computacional y trabajo colaborativo, entre otros [2]. Si bien la asignatura Informática también está incluida en el Ciclo básico del Consejo de Educación Técnico Profesional, pero con 2 horas semanales, este trabajo no hará referencia a este Consejo.

En el contexto indicado la edad de los alumnos y sus intereses se transforma en una variable fundamental a la hora de planificar las actividades de enseñanza. Es así que resulta necesario realizar actividades que resulten motivantes y así fomentar el interés en los contenidos y la actividad [3]. Si bien es necesario apelar a recursos y tareas que motiven al estudiante, no se debe olvidar que la institución educativa tiene un objetivo esencialmente académico, por lo que la motivación no será un fin, sino que un medio para lograr y lograr aprendizajes. Dicho de otro modo, toda actividad que motive a los estudiantes se justifica en tanto y cuanto la misma coadyuve a generar aprendizajes académicos; debiéndose tener en cuenta que puede ser un fin académico la adquisición de habilidades actitudinales.

Es así que uno de los desafíos con los que se debe enfrentar todo docente es la construcción de secuencias didácticas [4] que resulten adecuadas para lograr aprendizajes en los estudiantes, siendo un caso particular aquellas a realizar cuando el contenido a enseñar está asociado a la programación. En este contexto el docente debe tomar decisiones didácticas [3], las cuales consisten entre otras cosas en optar por cómo trabajar los contenidos fundamentales de programación y la construcción de actividades adecuadas. Para el caso de los contenidos disciplinares el docente deberá tener en cuenta las características del mismo, su relación con otros saberes y la integración adecuada en un problema o actividad. Al mismo tiempo es fundamental que estas actividades, que se relacionan con la programación, resulten motivantes a los estudiantes, que fomenten que estos últimos quieran aprender y estudiar el contenido propuesto por el docente.

Las formas de motivar a los estudiantes son múltiples, pudiendo centrarse en la afiliación, el poder o el logro [3]. Destaca particularmente la motivación por el logro, donde la misma se asocia a que los estudiantes comprendan que pueden y son capaces de resolver los problemas planteados. Sin embargo lo anterior no es suficiente, pues previamente a lo dicho es necesario que el estudiante quiera realizar la actividad, involucrarse activamente en ella y no sea un mero receptor de saberes determinados por el docente. De este modo la propia consigna de la actividad presentada debe ser tal que genere curiosidad e interés por su resolución.

Una de las formas de lograr lo anterior, consiste en la realización de juegos en los cuales clase a clase se incorpora un nuevo concepto o procedimiento [5] y están orientados a emular un juego que ya existe [6]. Este tipo de actividades se ha visto que resulta interesante para los alumnos, captando su atención y permitiendo así dirigir los esfuerzos de los mismos. Para que la actividad resulte motivante es necesario conocer al grupo, hacer un buen diagnóstico inicial y adaptar el trabajo en función de la respuesta de los estudiantes. Al mismo tiempo es posible que el propio juego sea estructurante de la actividad, que le dan sentido y permita hacer que los contenidos y objetivos se integren de forma natural. Para lograr lo anterior es necesario planificar cuidadosamente la progresión del trabajo, la incorporación de nuevos contenidos y su relación con el juego en sí mismo. En caso de no lograrse la adecuada secuenciación de actividades, y contenidos, se corre el riesgo que la primera no sea otra cosa que una sumatoria de los segundos, coherentes desde la disciplina, pero no desde la actividad que se planifica.

Otro elemento a considerar es el lenguaje usado, Scratch en este caso, El lenguaje se muestra como un buen primer lenguaje de programación en un curso post primario, que permite el aprendizaje los conceptos fuertes de programación [7]. Al mismo tiempo Scratch se muestra como un lenguaje de fácil uso y aprendizaje, con sintaxis reducida, el cual permite la integración de conocimientos de distintas áreas o disciplinas [8]. Las fortalezas de Scratch son tales que permite la construcción de programas que se integran con creciente complejidad [9]. Por último decir que Scratch destaca como un lenguaje de programación que permite el desarrollo de competencias que además de aplicar a la informática, también lo hace para otras asociadas a la comunicación o el lenguaje [10], en particular cuando las mismas se relacionan a actividades lúdicas [11].

II. SECUENCIA DE ACTIVIDADES

Se presenta una secuencia de actividades donde se trabajan contenidos de programación en Scratch para estudiantes de la asignatura Taller de Informática de Ciclo básico de la Dirección General de Enseñanza Secundaria. En nuestro caso se opta planificar la unidad didáctica de programación en función de tres actividades principales, juegos, sobre los cuales se articulan los contenidos programáticos.

Cada juego se desarrolla durante tres clases de dos horas pedagógicas de 45 minutos, seis horas en total. La primera hora de cada clase se orienta al trabajo con aquellos contenidos ya vistos por el alumno, que permiten ejercitar y profundizar los conocimientos adquiridos, siendo esto parte fundamental del proceso de aprendizaje [12]. La siguiente hora se usa para introducir los nuevos contenidos, para los cuales se procura que los estudiantes identifiquen su necesidad y pertinencia, buscándose generar situaciones de acción didáctica [13]. Dada la importancia de la computadora en el proceso de aprendizaje de la programación en enseñanza media y su didáctica [14], es que la presentación, análisis y trabajo con los nuevos contenidos se hace con uso de la misma; y no desde el análisis de soluciones en las cuales las mismas no se prueban y analizan sin el uso de la computadora. A su vez el primer día de cada actividad (que como dijimos está compuesta por un total de 6 horas pedagógicas), se presenta el juego, analiza sus características y grandes elementos constitutivos, procediendo a una descomposición del problema (objetos y las relaciones entre los mismo, eventos, condiciones de victoria o finalización, etc.).

La solución al juego propuesto por el docente, que es trabajada en las clases, no exige que algunos estudiantes incorporen nuevas sentencias a la solución (como sonidos o un manejo más intensivo de los disfraces) u opten por hacer modificaciones personales que entienden personalizan y mejoran la programación del juego (como hacer énfasis en el manejo de eventos para mover los objetos o en el diseño de los mismos, por ejemplo). Debe tenerse en cuenta que la dificultad de cada uno de los juegos es ligeramente superior al anterior, por lo que la aplicación de contenidos en la primer hora de cada clase implica necesariamente la resolución de problemas más complejos que el anterior.

Tabla I muestra los juegos usados: la descripción de los mismos, los principales contenidos y la descripción de qué objetos y acciones se programa en cada una de las clases de las tres asociadas para cada juego.

TABLA I JUEGOS, CONTENIDOS Y CLASES PARA SU SOLUCIÓN

Nombre y descripción de juego	Contenidos	Clases y programación
<p>Bosque encantado Un caballero (jugador) debe llegar a su castillo atravesando un bosque lleno de enemigos. Caballero: su objetivo es llegar al castillo sin ser atrapado por los enemigos, de ser así el juego termina (PERDER). Únicamente puede moverse a la izquierda o derecha. Fantasmas: se mueven verticalmente rebotando en bordes Bruja: tiene un movimiento “diagonal” rebotando en bordes</p>	<p>Estructura de repetición por siempre, estructura de selección simple. Condiciones y eventos</p>	<p>Programar enemigos: “fantasma” y “bruja” Programar “caballero” (controles) Programar interacciones y finales</p>
<p>Lluvia de meteoritos El jugador controla una NAVE que se mueve en la parte inferior de la pantalla con el objetivo de resistir el mayor tiempo posible a una lluvia de meteoritos. El movimiento de la nave se controla de izquierda a derecha solamente. Los meteoritos aparecen en la parte superior de la pantalla aleatoriamente y mueven hacia abajo. Si un meteorito golpea la nave se pierde una vida. El juego termina cuando ya no quedan vidas. Un contador de tiempo inicia al comenzar un juego y el jugador cuenta con 3 vidas.</p>	<p>Variables para la construcción de contadores (en este caso VIDAS) Saberes previos: Estructura de repetición por siempre, estructura de selección simple, Condiciones Eventos</p>	<p>Programar “nave” Programar “meteoritos” e interacciones Programar final de juego, contador de vidas</p>
<p>Tiro al blanco El jugador controla el lanzamiento de un dardo. El objetivo del juego es explotar la mayor cantidad de globos en 1 minuto. El juego finaliza luego de terminado ese tiempo. El dardo se lanza desde la izquierda de la pantalla al presionar una tecla. Antes de ser lanzado se mueve continuamente en forma vertical dificultando en tiro. En caso de darle a un globo se suma un punto, luego de finalizado el tiro el dardo vuelve a la posición inicial. El gato suelta los globos continuamente que pueden subir hasta la parte superior de la pantalla o explotar por el dardo.</p>	<p>Repetir hasta que (condición) Saberes previos: Estructura de repetición por siempre, estructura de selección simple, Condiciones Eventos Variables.</p>	<p>Programar “dardo” Programar “globo” Programar final de juego, contador de puntos.</p>

Figura 1 muestra las tres pantalla de presentación de los juegos indicados y cuyo diseño de escenario es proporcionado por el docente. Obsérvese lo dicho previamente en el sentido que algunos alumnos podrán verse tentados en modificar el diseño de meteoritos, el movimiento de un fantasma o modificar el objeto que suelta los globos (el gato de Scratch) del juego Tiro al blanco.

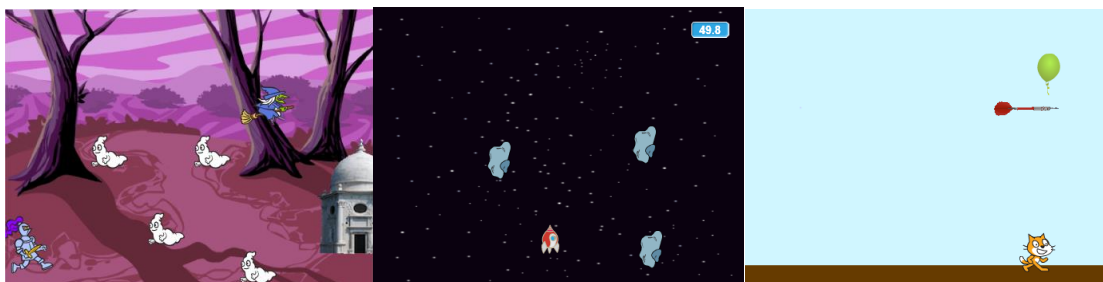


Figura 1. Pantalla de presentación del juego Bosque encantado, Lluvia de meteoritos y Tiro al blanco respectivamente.

III. CONCLUSIONES

Si bien la secuencia presentada no se ha comparado con otras en las cuales el juego no se considere como elemento constitutivo y estructurante, se está en condiciones de afirmar que la misma es vista como positiva por los alumnos. En efecto, se ha percibido la motivación de los estudiantes, su interés por la actividad y el deseo de resolverla. Al mismo tiempo se ha constatado que los estudiantes han logrado resolver las situaciones problemas. Por lo expuesto se entiende viable el trabajo con actividades como las planteadas, debiendo ser especialmente cuidadosos en la correcta secuenciación y progresión de contenidos.

IV. REFERENCIAS BIBLIOGRÁFICAS

- [1] P. J. Denning, "Is computer science science?" en *Communications of the ACM*, 48(4), 27, 2005
- [2] Dirección General de Enseñanza Secundaria. Planes de asignaturas. Recuperado de <https://www.ces.edu.uy/index.php/propuesta-educativa/20234>, 2016.
- [3] E. Fiore y J. Leymonié. Didáctica práctica para Enseñanza Media y Superior. Montevideo, Editorial Grupo Magro, 2007.
- [4] S. Antúnez, L. del Carmen, F. Imbernon, S. Parcerisa y A. Zabala. Del proyecto educativo a la programación de aula. España, Editorial GRAÓ, 1992.
- [5] A. Miños, "Primer curso de programación en Enseñanza Media: la experiencia con scratch" en *Didasc@lia: Didáctica y Educación*, 5(1), 2014, pp. 47 – 54.
- [6] R. Muñoz et al., "Uso de Scratch y Lego Mindstorms como apoyo a la docencia en Fundamentos de programación" en *Actas de las XXI Jornadas de la Enseñanza Universitaria de la Informática*, Universitat Oberta La Salle, Julio de 2015, pp. 248 – 254.
- [7] U- Wolz, H. Leitner, D. Malan y J. Maloney, "Starting with scratch in CS 1" en *Proceedings of the 40th ACM Technical Symposium on Computer Science Education - SIGCSE '09*, 2009
- [8] J. Maloney, M. Resnick, N. Rusk, B. Silverman y E. Eastmond, "The scratch programming language and environment" en *ACM Transactions on Computing Education (TOCE)*, 10(4), 16, 2010.
- [9] M. Resnick et al., "Scratch: programming for all" en *Communications of the ACM*, 52(11), pp. 60 – 67, 2009.
- [10] C. Scaffidi y C. Chambers, "Skill progression demonstrated by users in the Scratch animation environment" en *International Journal of Human-Computer Interaction*, 28(6), pp. 383 – 398, 2012.
- [11] E. Vázquez-Cano, y D. Delgado, "La creación de videojuegos con Scratch en Educación Secundaria" en *Communication papers*, 4(06), pp. 63 – 73, 2015.
- [12] N. Talyzina, "The Computerization of Learning – Scientific Foundation" en *Soviet Psychology*, 27(1), pp. 79 – 88, 19889.
- [13] G. Brousseau, "La théorie des situations didactiques" en *Cours donné lors de l'attribution à Guy Brousseau du titre de Docteur Honoris Causa de de Universidad de Montréal*, Montréal, Canadá, 1998.
- [14] A. Miños, "Elementos estructurantes de la Didáctica de la Informática" en *Virtualidad, Educación y Ciencia*, 8(14), 2017, pp. 100 – 110.