

**PRIMER CURSO DE PROGRAMACIÓN EN ENSEÑANZA MEDIA: LA EXPERIENCIA CON SCRATCH**

PRIMER CURSO DE PROGRAMACIÓN CON SCRATCH

AUTOR: Alejandro Miños Fayad<sup>1</sup>DIRECCIÓN PARA CORRESPONDENCIA: Instituto Normal de Enseñanza Técnica. Montevideo, Uruguay. E-mail: [alejandromifa@gmail.com](mailto:alejandromifa@gmail.com)

Fecha de recepción: 22 - 03 - 2013

Fecha de aceptación: 10 - 12 - 2013

**RESUMEN**

En los centros de educación media el concepto de informática ha mutado, pasando de ser la enseñanza de herramientas de ofimática, a la enseñanza de programación (aunque tímidamente). La selección del primer lenguaje de programación es de vital importancia, pues determina el énfasis que tendrá el curso. En efecto, el lenguaje de programación determinará el énfasis que pondrá el docente en la sintaxis del lenguaje, pudiendo ser reducida esta dificultad. Al mismo tiempo debemos procurar que los programas realizados sean atractivos para los alumnos, no sólo en lo relacionado con aspectos visuales, sino que también en su utilidad. Scratch brinda la posibilidad de conjugar aspectos visuales atractivos con una sintaxis minimalista, pudiendo hacer énfasis en aspectos conceptuales y operativos más que sintácticos. El presente documento recoge la experiencia relacionada con la enseñanza de la programación en un primer curso, procurando simplificar al máximo los aspectos sintácticos, sin reducir el esfuerzo conceptual y operativo de los alumnos.

PALABRAS CLAVE: iniciación a la programación; experiencia; Scratch.

**FIRST YEAR PROGRAMMING IN MEDIA EDUCATION: EXPERIENCE WITH SCRATCH****ABSTRACT**

In secondary education centers the concept of computing has mutated from being teaching office tools, teaching programming (albeit timidly). The choice of first programming language is critical because it determines the emphasis to be on course. Indeed, the programming language that will determine the emphasis on teaching the language syntax, this difficulty can be reduced. At the same time we must ensure that the programs implemented are attractive to students, not only in relation to visual aspects, but also in their usefulness. Scratch offers the possibility of combining attractive visuals with a minimalist syntax and can emphasize conceptual and operational aspects rather than syntactic. This paper describes the experiences related to the teaching of programming in

---

<sup>1</sup> Profesor de Informática. Postgrado en Estrategia. Docente de asignaturas técnicas del área informática y de Didáctica de la Informática.

a first course, trying to simplify the syntactic aspects, without reducing the conceptual and operational effort of the students.

KEYWORDS: introduction to programming; experience; Scratch

## INTRODUCCIÓN

En el Instituto Normal de Enseñanza Técnica, de Montevideo – Uruguay, se dicta el profesorado de informática, el cual permite obtener un título docente en educación media para la especialidad Informática. Durante dos años se realiza la práctica docente con un profesor tutor, la correspondiente al último año de la carrera está a cargo del estudiante, siendo de carácter anual.

Durante el año 2011 realizamos la práctica docente de 4º año, del profesorado de informática en el Liceo 12 “Congregado de Tres Cruces”, centro educativo de enseñanza media básica (de primer a tercer año post primarios). Como parte del programa a dictar en la asignatura Informática se incluye una introducción a programación, usando Scratch.

La enseñanza de la Programación en Ciclo Básico de enseñanza secundaria, implica todo un desafío para los docentes y profesores de Enseñanza Media. El solo hecho de lograr aplicar estrategias de motivación no es una tarea trivial, y requiere un profundo análisis de los intereses y conocimientos previos de los alumnos. Al mismo tiempo debemos generar las condiciones para que el alumno se apropie del conocimiento, tarea que implica acercarnos a la estructura cognitiva del estudiante, logrando así tener un conocimiento de sus posibilidades de abstracción y entendimiento de los problemas y estrategias propuestas.

Es así que la pregunta que cabe es siempre la misma: ¿es significativo para los alumnos los ejemplos y actividades que planteamos?, ¿logramos presentar el contenido de forma suficientemente interesante como para captar la atención e interés de los alumnos?

Seguramente estamos de acuerdo con que trabajar sobre una liquidación de sueldos no representa mucho para los alumnos de Ciclo Básico, así como que el uso excesivo de ejemplos matemáticos no es lo más adecuado. En efecto, trabajar sobre el área de un polígono, un promedio, máximos o mínimos numéricos, aunque resulte útil para presentar los conceptos y procedimientos programáticos a tratar, puede resultar irrelevante para los alumnos. Es así que se hace necesario presentar actividades cercanas a la realidad de los estudiantes, de modo que estos perciban el nuevo conocimiento como propio.

El presente artículo presenta la experiencia relacionada con la enseñanza de la programación, en un curso de enseñanza media y básica.

## DESARROLLO

El programa de Informática de segundo año de Ciclo Básico consta de dos grandes áreas curriculares, la primera compuesta por la operación de

programas clasificados dentro de la ofimática, y la segunda que abarca una introducción a la programación estructurada. Para trabajar con este último eje temático se lo hace con un software llamado Scratch, el cual forma parte de los programas que vienen instalados con las computadoras XO<sup>2</sup>. Pero aunque la programación forma parte de la estructura curricular del curso, y al no existir inspección docente en la asignatura, no existen pautas metodológicas sobre cómo se debe trabajar esta unidad programática, lo que hace que sea el docente quien defina los criterios y lineamientos metodológicos en base a su experiencia.

Para poder programar es necesario conocer y manejar ciertos conceptos y procedimientos, propios de la ciencia de la Computación. De este modo, todo proceso de enseñanza – aprendizaje debe procurar alentar la adquisición de ciertas habilidades como ser: la posibilidad de formalizar un problema más allá de las ambigüedades del lenguaje natural, la capacidad de reconocer la existencia de estructuras iterativas presentes en el lenguaje natural y la posibilidad de dividir un problema en distintas partes mediante la estrategia de divide y vencerás, de modo que la suma de cada una de ellas dé como resultado el problema original, entre otras.

Pero el proceso anterior, de tinte claramente formal, se debe aprender de a poco, no siendo instantáneo, trivial ni mera inspiración. Cuando los estudiantes comienzan a programar debemos considerar si haremos énfasis en aspectos teóricos sobre lo que es una buena programación, o debemos fomentar la creatividad y el análisis algorítmico que se realiza. Si optamos por la segunda estrategia debemos tener en cuenta que existen problemas, que incluso desde una visión formal, admiten varias soluciones.

Nuestra experiencia docente nos ha mostrado que con frecuencia los procesos excesivamente formales, aunque necesarios, son percibidos como agobiantes por los alumnos, los cuales no relacionan claramente los aspectos sintácticos o teóricos con su aplicación.

El problema de la elección del primer lenguaje nos lleva a la necesidad que las estructuras de control deben poder generalizarse fácilmente, reducir al máximo la sintaxis procurando minimizar el uso de delimitadores de bloques, y con una notación sencilla. Python es una opción como primer lenguaje para aprender a programar, que se está abriendo camino en el mundo académico, pero no resulta atractivo el uso de sentencias escritas por parte de los alumnos.

Scratch brinda la posibilidad de programar sin esa estructura que es percibida como excesivamente rígida, la cual impregna los libros de texto. Si bien existe una clara sintaxis de las sentencias esta es percibida como relativamente fácil, se puede trabajar sin preocuparse demasiado por ella, dentro de ciertos márgenes de maniobra, dejando que el entorno gráfico haga buena parte de nuestro trabajo (Figura 1).

---

<sup>2</sup>El Plan Ceibal es un plan de inclusión social y tecnológica caracterizado por otorgar a cada alumno de centros educativos públicos una laptop (llamada XO). El plan es desarrollado en la República Oriental del Uruguay.

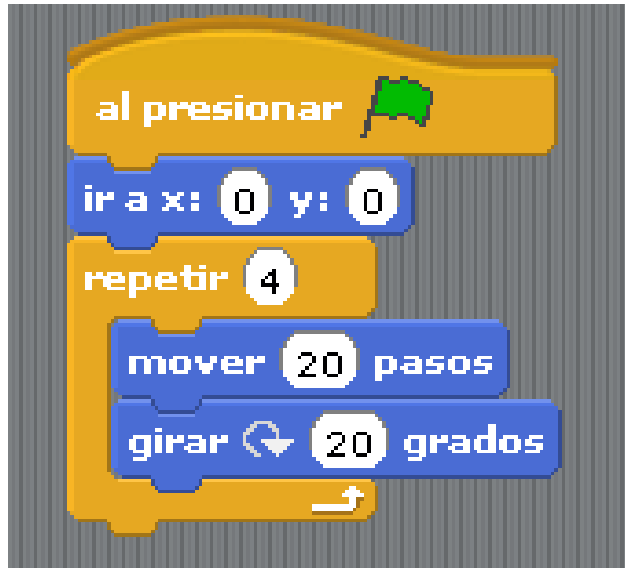


Figura 1. Entorno gráfico de Scratch.

Pero la sintaxis no es el único problema al cual nos enfrentamos desde la programación. El uso de acumuladores, contadores y estructuras de control no es trivial, y requiere que el alumno aprehenda, tome, se apropie de su concepto e idea de uso. Quién sabe porqué, pero quienes nos dedicamos a la enseñanza de la programación, tendemos a recurrir a ejemplos relacionados con las matemáticas para trabajar con los conceptos anteriores. Es así que si queremos trabajar con un acumulador podemos recurrir al ejemplo en el cual dados  $n$  números determinamos cuantos son mayores o menores de otro dado, o promediar un conjunto de valores numéricos, por ejemplo. Es claro que no siempre, pero sí es común recurrir a este tipo de ejemplos.

Lo anterior nos lleva a lo ya mencionado, y que se reduce a las preguntas que más de una vez hemos oído por parte de nuestros alumnos, y que en ocasiones nos resulta difícil contestar:

“¿Y para que quiero saber el promedio de esos números?”

“¿Para qué quiero saber si un número es positivo o negativo?”

“¿De qué sirve sumar solamente los números que son mayores de 10?”

Durante la práctica docente, decidimos renunciar a la clásica estrategia basada en la presentación de sintaxis, su uso controlado y posterior aplicación. En lugar de lo anterior, y conscientes de las dificultades que implica ello, decidimos trabajar enfocados en pequeños “proyectos”, dando pautas sobre el trabajo a realizar, enfocados en los efectos visibles por los estudiantes. De este modo, buscábamos que los estudiantes adquirieran un grupo de conocimientos generales y comunes a todos, mientras les dábamos la posibilidad de investigar por su cuenta, considerando sus intereses.

Pero al mismo tiempo renunciamos a recurrir a los mismos ejemplos matemáticos que hemos referenciado. En ningún momento propusimos

promediar valores numéricos, obtener el mayor de una lista o sumar, restar o realizar operaciones matemáticas.

Desde nuestra perspectiva, consideramos que lo más importante es que el alumno se apropie de las estructuras básicas de la programación, y en este nivel un uso limitado de variables. Es así que merced a los beneficios que nos proporciona el entorno gráfico, pudimos trabajar en la interacción que brindan los objetos con el usuario.

Siendo el objetivo introducir a los estudiantes en los conceptos de programación, operativamente esto fue reflejado en la realización de un video juego, el cual debía ser controlado por los estudiantes.

El proyecto fue dividido en una serie de etapas las cuales tenían por objetivo:

1. Presentar la sintaxis necesaria para mover un objeto.

En esta primera parte nos centramos en el trabajo con estructuras repetitivas, el uso del lápiz y el trabajo con coordenadas. Procuramos que los alumnos, mediante la realización de actividades controladas, sientan lentamente la necesidad de suplantar la reiteración de código de programación por estructuras iterativas.

2. Poder editar un objeto y el fondo.

Casi concurrentemente con lo anterior nos abocamos a brindar un entorno de trabajo agradable a los alumnos, logrando que estos lo personalicen.

Debemos recordar que uno de los problemas a los cuales nos enfrentamos en la programación, es que tendemos a centrarnos en la resolución de problemas sin interacción con el usuario, dejando esto para más adelante. Esa forma de trabajo genera como inconveniente el hecho que si bien el programa resuelve los problemas, no resulta motivante para el alumno desde el punto de vista gráfico.

3. Controlar interactivamente el objeto, logrando que este se mueva a voluntad del usuario.

No solo es importante la existencia de objetos que resulten interesantes para el alumno, sino que más importante es que los mismos puedan ser controlados por el alumno – programador – usuario.

Pero la interactividad de los objetos, además de procurar ser motivante, tuvo por objetivo el trabajo con la estructura de control SI, la que es fundamental en la programación. De este modo logramos que los alumnos no perciban la nueva sentencia como “una más”, sino que se genere la necesidad de ella, de su uso, y que su utilidad sea identificada claramente.

El uso de variables estuvo orientado a lograr la interactividad de programa realizado. Dado que las variables usadas fueron de uso

limitado, es que nos preocupamos principalmente en identificar el comportamiento de las variables (lo que es llamado roles) más que sus características y definición.

4. Mejorar la interactividad de los objetos y el entorno, generando efectos como los choques, sonidos y condiciones de victoria.

Esta última etapa se caracterizó por la sumatoria de las etapas anteriores, la obtención del producto definitivo y el refinamiento sucesivo en el uso del lenguaje técnico. En efecto, además de obtener un video juego, incorporamos conceptos como programación concurrente, multiusuario o animación.

La estructura de trabajo anterior permitió que los estudiantes visualizaran sus actividades desde una perspectiva lúdica, por sucesivas aproximaciones a la solución deseada, trabajando en grupos o individualmente y cooperando entre pares.

Al mismo tiempo, y conceptualmente lo más importante, se logro de que los estudiantes se apropien de los conceptos de:

1. Sentencias y bloques se sentencias.
2. Uso limitado de contadores y acumuladores.
3. Iteraciones y condiciones.
4. Conceptos generales de programación concurrente.

## CONCLUSIONES

En general, los alumnos se mostraron satisfechos con el trabajo realizado. En efecto, alumnos que se habían mostrado relativamente apáticos en relación al resto de las actividades, se mostraron entusiasmados por las actividades de programación.

El trabajo con objetos que podían ser modificados por ellos (casas, autos, animales o personas), y luego controlados, resultó motivante, dedicándose muy esmeradamente a ello. Al mismo tiempo, la incorporación paulatina de términos técnicos no generó inconvenientes a los estudiantes, quienes las incorporaron paulatinamente a su vocabulario.

En efecto, el trabajo con un entorno que resulte altamente interactivo para los alumnos se ha mostrado como eficaz, siendo “natural” la asimilación de nuevos conceptos.

Como aspecto a corregir debemos considerar que no ha resultado trivial la diferenciación entre el concepto y el efecto visual que ese concepto genera. Es así que para trabajar con la sentencia SI emulamos un objeto que choca contra una pared. El alumno debía, y lo logró, verificar que se hacía necesaria la evaluación de una condición (la superposición de dos objetos) para que un objeto “choque” contra otro. No resultó trivial, al principio, que los alumnos separasen el concepto de condición y sentencia SI del de “choque” entre objetos.

Del mismo modo debemos tener en cuenta que la realización de un video juego, eje de toda la unidad, supone un conocimiento relativamente profundo de la herramienta y la programación, lo cual no se daba en los estudiantes. Se debió pautar cada actividad lo suficientemente bien, cosa para nada trivial, como para que el alumno se retire del aula con la sensación de haber hecho parte del juego.

El uso de herramientas visuales de programación como Scratch permite:

1. Un mejor análisis del código existente, así como su interpretación.
2. Reduce la dificultad asociada con la sintaxis.
3. Permiten trabajar con algoritmos que no hacen uso de ejemplos matemáticos.

Es así que podemos decir que Scratch es una buena herramienta para aproximar a los estudiantes a la programación, reduciendo la brecha existente entre el algoritmo como solución de un problema y la interacción que supone su uso con el alumno – programador – estudiante.

No obstante lo anterior, y para terminar, debemos considerar no caer en el error de no asociar los resultados visuales obtenidos, con los conceptos tratados, recordando que los primeros no son más que el efecto, en tanto que los segundo la causa (y concepto) que debemos trabajar.

#### BIBLIOGRAFÍA

Dowek G. (2005) – ¿Qué informática enseñar en el liceo?, École polytechnique. Disponible en: <http://www.informate.edu.uy/Textos/dowek.pdf> [consultado por última vez el 21 de marzo de 2013]

Marzal A. – Llorens D. – García I., Aprender a programar con Python: una experiencia docente, Universitat Jaume I. Disponible en: <http://es.tldp.org/Presentaciones/200309hispalinux/15/15.pdf> [consultado por última vez el 21 de marzo de 2013]

Pérez M. y otros, Evaluación del Plan Ceibal 2010. Documento resumen, Administración Nacional de Educación Pública, Uruguay.

Saavedra J. – Silveira A., Algunas dificultades en el aprendizaje del concepto de variable, Reporte Técnico RT 11-07 (PEDECIBA Informática – Instituto de Computación – Facultad de Ingeniería – Universidad de la República), 2011, 1 – 9. Disponible en: <http://www.fing.edu.uy/inco/pedeciba/bibliote/reptec/TR1107.pdf> [consultado por última vez el 21 de marzo de 2013]

Villalobos J., Proyecto CUIP2 – Una solución integral al problema de enseñar y aprender a programar, 10° Premio Colombiano en Investigación Educativa (Departamento de Ingeniería de Sistemas y Computación – Facultad de Ingeniería – Universidad de Los Andes), 2009, 1 – 39. [http://www.colombiaaprende.edu.co/html/mediateca/1607/articles-205832\\_recurso\\_1.pdf](http://www.colombiaaprende.edu.co/html/mediateca/1607/articles-205832_recurso_1.pdf) [consultado por última vez el 21 de marzo de 2013]

