

## Programación visual en la formación de docentes de Biología

*Gimena Betina Fussero<sup>1</sup>, Marcela Cristina Chiarani<sup>2</sup>.*

*<sup>1</sup>Departamento de Enseñanza. Facultad de Ciencias Exactas, Físicas y Naturales. Universidad Nacional de Córdoba. Av. Vélez Sarsfield 299. Córdoba Capital. Argentina.*

*<sup>2</sup>Departamento de Informática. Facultad de Ciencias Físico, Matemáticas y Naturales. Universidad Nacional de San Luis. Ejército de los Andes 950. San Luis. Argentina.*

*<sup>1</sup>gimefussero@gmail.com, <sup>2</sup>mcchiarani@gmail.com*

### Resumen

Se presenta una investigación acerca de una experiencia de formación docente continua sobre la Programación como herramienta de Enseñanza de Biología. Participaron profesores de escuelas secundarias de la ciudad de Córdoba y alumnos del Profesorado en Ciencias Biológicas de la Facultad de Ciencias Exactas, Físicas y Naturales de la Universidad Nacional de Córdoba (Argentina). La metodología utilizada fue taller ya que propicia la participación activa del alumno dentro del proceso de aprendizaje. La propuesta del mismo se centró en aplicar el pensamiento creativo en conceptos biológicos a fin de desarrollar y remixar proyectos. El entorno de programación utilizado fue Scratch ya que cuenta con diferentes ventajas, por ejemplo: es un lenguaje visual, no requiere conocimientos previos, se encuentra disponible en la web y como aplicación de escritorio. La estrategia de evaluación de los aprendizajes empleada fue continua con una instancia de evaluación final. Los resultados indican que es factible utilizar el entorno de Programación Scratch para desarrollar materiales educativos en y para la Enseñanza de la Biología.

**Palabras clave:** Biología, Enseñanza, Programación, Scratch, Formación docente.

### Introducción

Las Tecnologías de la Información y la Comunicación (TIC) tienen consigo diferentes potencialidades. Respecto a la Programación, permiten experimentar en todas sus etapas: análisis, diseño, desarrollo y prueba promoviendo el desarrollo de capacidades y competencias. En particular los entornos de Programación que presentan un lenguaje visual pueden ser empleados como recursos educativos. En sus dos perspectivas, una para el desarrollo de proyectos por parte de los alumnos y la otra para el desarrollo de material didáctico por parte del profesor.

Si bien para los docentes de Educación Tecnológica la Programación de computadoras es un medio de creación y expresión. Cabe preguntarse sobre los docentes de disciplinas como por ejemplo, la Biología ¿Deberían aprender a programar para desarrollar material didáctico digital? ¿Podrían utilizar la Programación como un recurso didáctico en sus clases? Si se analizan las habilidades del siglo XXI que deben adquirir los profesores y si a la vez se indaga en las áreas de competencias digitales requeridas actualmente según la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO,

2008) (información, comunicación, creación de contenido, seguridad y resolución de problemas) se desprende que tanto la tercer habilidad (creación de contenido) como la última de ellas (resolución de problemas) impulsan, por ejemplo, el desarrollo de materiales didácticos utilizando creativamente la Tecnología. Sin duda, el desafío es propiciar que los docentes de Biología desarrollen dichas habilidades en su formación continua.

En función de ello, en este trabajo se presenta el diseño y análisis del taller "Imagina, Crea y Comparte con Scratch" que fue destinado a profesores de escuelas secundarias y alumnos del profesorado de dicha casa de estudios. Este taller tuvo lugar en el marco de la III Jornadas de Investigación Educativa y II Jornadas de Práctica de la Enseñanza del Profesorado en Ciencias Biológicas de la Facultad de Ciencias Exactas, Físicas y Naturales de la Universidad Nacional de Córdoba.

## Referentes Teóricos

La Programación de computadoras es el arte de conseguir que una computadora haga lo que se le indica. Como resultado se obtiene un programa informático (*software*), que simplemente es un conjunto de instrucciones que le señala a una computadora cómo realizar una tarea específica. A modo de ejemplo, se podría señalar que un programa informático es como una receta de cocina, es un conjunto de instrucciones que explica cómo hacer algo de forma clara, precisa y no ambigua.

El año 1967 Seymour Papert desde el Laboratorio de Inteligencia Artificial del Instituto Tecnológico de Massachussets (MIT) creó con su equipo, el lenguaje de Programación conocido como Logo, con un fin educativo. En la Argentina llegó a las escuelas primarias y secundarias en los años 80. En muchas de ellas se implementaron talleres para la Enseñanza de la Programación, como asignatura extracurricular, para promover el pensamiento creativo y la exploración de ideas.

En estos últimos años resurge la propuesta de enseñar informática a todos, bajo la propuesta del pensamiento computacional. Aprender a programar o desarrollar el pensamiento computacional como sugiere Jeannette Wing (2006) es una habilidad que debe ser aprendida por todos los niños, al igual que leer, escribir y aritmética. El potencial de aprender a programar es promover el desarrollo de habilidades para resolver problemas informáticos, reflexionar sobre las soluciones planteadas y promover la búsqueda de errores con el fin de optimizar la propuesta. Resnik refuerza esta idea cuando expresa que "*La habilidad para programar permite a las personas "escribir" nuevos tipos de cosas, tales como: Historias interactivas, juegos, animaciones y simulaciones*" (Resnik, 2013, p. 3).

En línea con lo expresado, se observa que para Popper, "*El proceso de aprender consiste principalmente en correcciones a expectativas que no se cumplen*" (Popper, 1985, p. 156) y que son justamente, las mismas que desencadenan el proceso de ensayo y error que se promueve a través de la Enseñanza de la Programación de computadoras, y de la solución de problemas mediante el uso de computadoras. Es significativo aclarar,

que aprender a programar computadoras NO es aprender un lenguaje de programación y su sintaxis.

El estudiante debe concebir, entonces, al aprendizaje como la posibilidad de experimentar, equivocarse y aprender. Sin duda la Enseñanza de la Programación coincide con lo expresado por Freire cuando señala que "*Solo existe saber en la invención, en la reinvención, en la búsqueda inquieta e impaciente*" (Freire, 2005, p.79).

#### *Entorno de programación visual Scratch*

Al momento de seleccionar un lenguaje de programación, se debe tener en cuenta al grupo de alumnos y sus conocimientos previos.

Scratch fue desarrollado en el *Lifelong Kindergarten del Media Laboratory* del MIT (Universidad de California, EE.UU). Es una herramienta de acceso libre y gratuita, se puede descargar o utilizar en forma online desde la página oficial. Revisando su sitio web, se observa que al aprender a programar con Scratch se aprenden importantes estrategias para resolver problemas, diseñar proyectos y comunicar ideas. También se expone que su uso permite el desarrollo del pensamiento lógico, fomenta la creatividad y facilita el pensamiento sistémico. Todo ello posibilita una experiencia educativa enriquecedora.

Otra de las ventajas que posee es que cuenta con una interface amigable. En ella la programación está basada en bloques de diferentes colores, con el fin de generar una secuencia de acciones, que deben seguir los personajes, en escenarios preestablecidos o creados. Se pueden producir juegos y animaciones. Además, en el sitio web de Scratch, los proyectos pueden ser compartidos, para inspeccionar su código, aprender y reutilizarlos.

El software Scratch está basado en la teoría del conocimiento constructivista, la cual expresa la necesidad de ofrecer a los alumnos herramientas que le posibiliten crear sus propios procedimientos para resolver una situación problemática, lo que implica que sus ideas se modifiquen y continúen aprendiendo. En esta instancia el papel creativo y constructivo de los errores, puede ser utilizado como una estrategia de aprendizaje. Los errores de Programación, son una simple (a veces... no tan simple) diferencia entre lo que se quería lograr y lo que realmente se ha logrado.

### **Propuesta De Formación**

La propuesta del taller se centró en utilizar el entorno de Programación Scratch aplicado al planteamiento y al desarrollo de diversos temas a través de la creación de proyectos para el área de Biología. Asimismo, se trabajó el concepto de compartir proyectos de elaboración propia, como la idea de reutilizar y combinar proyectos disponibles en la comunidad web de Scratch.

El temario desarrollado abarcó una introducción al pensamiento computacional con la práctica en el entorno gráfico de Programación para realizar los primeros pasos con Scratch. Se incluyó un desafío de crear nuevos proyectos y el de explorar y remixar

proyectos existentes. Además hubo instancias para compartir dudas y reflexionar sobre los avances personales.

La metodología empleada fue la participación activa del alumno dentro del proceso de aprendizaje, a fin de ayudar a generar las competencias necesarias para apropiarse del Espiral del Pensamiento Creativo, propuesto por Mitch Resnick (2007) que sugiere como pasos: imaginar, crear, jugar, compartir y reflexionar.

Como desafío inicial se planteó crear un pequeño proyecto en el cual el gato (objeto) realizara movimientos simples, luego pudiera moverse dando el efecto que camina y regrese al mismo lugar. Para avanzar se les dio a los alumnos otro desafío, que les permitiera apropiarse de otros conceptos computacionales. Como una actividad más compleja se les propuso explorar proyectos existentes relacionados a Genética y que analizaran las acciones que ejecuta cada objeto para luego plantear posibles modificaciones. De igual forma, se propusieron dos instancias para compartir las dudas y logros alcanzados con todos los alumnos del curso.



A modo de ejemplo se muestra uno de los desafíos: Se proponía realizar un proyecto sencillo en el que la pelota rebota libremente y si toca las columnas de colores emite un sonido (figura 1). Para ello debían seleccionar, de los objetos preestablecidos, el objeto *soccer ball*. Luego dibujar los objetos, objeto 1 (columna verde) y objeto 2 (columna amarilla). Como fondo debían seleccionar *brickwall* que se encuentra preestablecido. Con este desafío se abordaron los conceptos computacionales de secuencia, selección y repetición.

## Resultados

De los 23 participantes, un 60% se correspondía a docentes y el 40% restante eran estudiantes. El 52% se encontraba en la franja etaria de 20-30 años mientras que el porcentaje restante abarcaba un intervalo que iba desde los 30 a los 63 años. Para finalizar la caracterización de los asistentes al taller, se les consultó sobre sus años de docencia. Del porcentaje que respondió (64%), un 32% se encuentra en un intervalo de 0-5 años de docencia, un 16% entre 15-20 años y el restante 16% más de 20 años de actividad docente.

Cuando se les consultó sobre las motivaciones que los llevaron a inscribirse en esta actividad, el 52% destacó las potencialidades que brindan las TIC en el aula. Expresiones como *"Me gustan las posibilidades didácticas que ofrecen las TIC"*, *"El aprender a manejar un programa de este tipo ya que sabemos que las nuevas tecnologías son las nuevas herramientas en la educación"*, *"Conocer una herramienta nueva que puedo incorporar*

en mis clases” dan cuenta de ello. Por otro lado, un 32% hizo alusión a la actualización/capacitación. Respuestas como “Actualizarme en el trabajo”, “La idea de una formación y capacitación permanente” y “Me interesa seguir perfeccionándome, sobre todo en lo relacionado a las TIC” ejemplifican dicha postura. Con porcentajes iguales (8%) los participantes destacaron la falta de conocimiento sobre las TIC (“Desconozco la utilización de instrumentos tecnológicos como recursos didácticos y su uso para la motivación de los estudiantes”) o hicieron alusión a otras cuestiones no detalladas aquí.

En relación a la frecuencia de la utilización de las TIC en las clases, se observa que un 32% alude a que las usa “constantemente/frecuentemente” en sus clases; un 16% indica que “no las utiliza” porque, si bien poseen computadoras, no poseen conexión a Internet. Un 8% subraya que las utilizan “poco” porque no tienen un buen soporte técnico mientras que el 12% señala que “no las utiliza nunca”. El porcentaje restante (32%) se agrupan quienes no respondieron y quienes hicieron mención a otras cuestiones.

Cuando se realiza un paralelo entre lo que los asistentes conocían sobre programación antes y después del taller (figura 2), la categoría “suficiente” que en un principio era de un 8% paso a un 64%. Las respuestas que hacían mención a “poco” cambiaron de 52% a 36% respectivamente. Por último, la categoría “nada” que en un principio se encontraba representada con un 40% luego del taller no hubo nadie que optara por dicha respuesta.



Figura 2: Percepción de los asistentes sobre sus conocimientos sobre Programación antes y después del taller.

Para finalizar, del porcentaje que respondió a este interrogante (20%), un 100% indicó que recomendaría el taller a colegas. El mismo porcentaje (20%) fue el que entregó la evaluación final del taller. Cabe destacar que el resto de los asistentes al finalizar la jornada comentaron que por cuestiones de agenda (finalización del ciclo lectivo) no iban a completar la evaluación final. Este es un dato significativo de aclarar ya que el bajo porcentaje que acreditaron el taller no fue por cuestiones de dificultad en el uso de la herramienta sino cuestiones externas al mismo.

De los participantes que finalizaron la instancia evaluativa final (20%), las modificaciones que realizaron sobre los proyectos incluían cambios de escenario, de objetos, de sonidos entre otros. A modo de ejemplo mostramos uno de los trabajos remixados por

un asistente al taller (figura 3). En el mismo se detallan las modificaciones realizadas al proyecto original.



Figura 3: Trabajo remixado por uno de los asistentes al taller. Se detallan las modificaciones realizadas.

## Conclusiones

Teniendo en cuenta, tanto, los resultados de la experiencia como así también su evaluación final, se destaca que fue una instancia donde docentes y estudiantes pudieron explorar un nuevo entorno con amplias potencialidades en los procesos de enseñanza y de aprendizaje. Partiendo del total desconocimiento de la plataforma, se llegaron a remixar diferentes ítem de los proyectos originales. Además teniendo en cuenta que la mayoría de los asistentes (55%) utiliza las TIC en sus clases (las aplicaciones más utilizadas son Power Point, videos, simuladores, entre otros); el entorno de programación visual Scratch podría convertirse en una nueva herramienta a utilizar en las mismas ya que luego del taller, las categorías que hacen alusión a ideas sobre programación "suficiente" y "poco" juntas suman un total de 100% contra un 60% inicial. Por todo lo antes expuesto es intención a futuro ampliar el curso a otros profesorados.

## Referencias Bibliográficas

- Freire, P. (2005). *Pedagogía del oprimido*. (2º Ed.). México: Siglo XIX Editores.
- Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO). (2008). *Estándares de competencias en TIC para docentes*. Recuperado de [http://portal.unesco.org/es/ev.phpURL\\_ID=41553&URL\\_DO=DO\\_TOPIC&URL\\_SECTION=201.html](http://portal.unesco.org/es/ev.phpURL_ID=41553&URL_DO=DO_TOPIC&URL_SECTION=201.html). <http://cst.unesco-ci.org/sites/projects/cst/default.aspx>
- Popper, K. P. (1985). *Sociedad abierta, universo abierto* (Conversaciones con Franz Kreuzer). Madrid, España: Tecnos.
- Resnik, M. (2007). *Sembrando las semillas para una sociedad más creativa*. Laboratorio de Medios, MIT, Massachussets. Recuperado de <http://eduteka.icesi.edu.co/articulos/ScratchResnickCreatividad>
- Resnik, M. (2013). *Aprender a programar, programar para aprender*. Laboratorio de Medios, MIT, Massachussets. Recuperado de <http://eduteka.icesi.edu.co/articulos/codetolearn>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.