

## **Enseñanza de la programación en la formación docente: pautas pedagógicas desde una experiencia con maestros de primaria**

Natalia Monjelat<sup>1</sup>

<sup>1</sup> Instituto Rosario de Investigaciones en Ciencias de la Educación (IRICE.CONICET-UNR)  
Rosario, Argentina  
monjelat@[irice-conicet.gov.ar](mailto:irice-conicet.gov.ar)

**Resumen.** Este trabajo presenta un estudio exploratorio y descriptivo en torno al “Taller de programación 1” de una especialización docente en Ciencias de la Computación, del que participaron 43 docentes de primaria. Se realizó un análisis de las características de los proyectos finales realizados con Scratch así como de los informes presentados donde se incluyeron preguntas de reflexión sobre el proceso desarrollado, con el objetivo de informar futuras propuestas de formación en estas temáticas. Los resultados señalan una mayor cantidad de juegos que de animaciones y de proyectos con contenidos educativos en vez de meramente recreativos. Asimismo, en los juegos se despliegan mayor cantidad de bloques vinculados a distintos conceptos de programación y se observa una mayor cantidad de objetos que en las animaciones, donde sí se emplearon más escenarios. Para la resolución de situaciones problemáticas los participantes recurrieron en gran medida al trabajo colaborativo con docentes y pares y señalaron el empleo de múltiples recursos propios de la comunidad de Scratch, que aparece como una herramienta apropiada para introducir a los docentes de primaria en el mundo de la programación, permitiéndoles la creación de producciones vinculadas con su práctica docente y con contenidos educativos.

### **1 Introducción**

Actualmente, se observa un interés renovado por introducir la Programación en los distintos niveles y modalidades del sistema educativo. Para ello, la formación de maestros en estas temáticas resulta un punto crucial [1]. Aunque en una gran mayoría de países, esta temática se encuentra aún en sus fases iniciales y no hay un consenso claro sobre qué implica su enseñanza o cómo debe ser introducida [2], en los últimos años han surgido diferentes iniciativas que centran sus esfuerzos en el desarrollo profesional de docentes en el campo de las Ciencias de la Computación, así como múltiples recursos y dispositivos disponibles para aprender a programar.

Respecto a las iniciativas, en el contexto internacional se ha optado por formar a maestros de otras áreas temáticas para enseñar estos contenidos, a través de capacitaciones en servicio o programas de desarrollo profesional intensivo [3]. Sin embargo, en los últimos años se observa un alejamiento de este tipo de propuestas formativas tradicionales de solo uno o dos días, en pos de la generación de espacios en los que los maestros desempeñen un papel

activo en el aprendizaje, aunque se observa una escasez de estudios que aborden la "experiencia docente" en estos nuevos trayectos [4]. En el contexto latinoamericano, la realidad de cada país es diferente en cuanto a políticas educativas y procesos de implementación de estas temáticas [5], lo que hace necesario adecuar las propuestas a las características de cada contexto y a sus particularidades [6]. En nuestro país, Argentina, se lanzó en el año 2013 el proyecto "Program.ar" por parte de la Fundación Sadosky, que a partir de diversas acciones acerca a niños y jóvenes al aprendizaje de las Ciencias de la Computación (CC). Asimismo, el Consejo Federal de Educación señaló al aprendizaje de la programación de importancia estratégica para el Sistema Educativo Nacional durante la escolaridad obligatoria (Resolución CFE N° 263/15, 2015) afirmando la relevancia de estos contenidos para la formación de los niños y jóvenes. En este marco, recientemente se iniciaron acciones de formación docente en estos temas, de la mano de convenios entre Institutos de Formación Docente, Fundación Sadosky y universidades nacionales. De estos acuerdos surgieron especializaciones docentes de dos años de duración que serán desarrolladas en diferentes regiones del país<sup>1</sup>.

Respecto a las herramientas y dispositivos, cabe señalar que la introducción de la programación en la escuela data de 1960, a partir del programa Logo introducido como un marco para enseñar matemáticas [7]. Sin embargo, ese entusiasmo inicial decayó ya que la sintaxis de los lenguajes utilizados resultaba compleja para los niños, la programación era generalmente introducida con actividades desconectadas de sus intereses (generación de números primos, líneas simples de dibujo) y en contextos donde no se ofrecía guía frente a los errores, o no se promovía mayor exploración cuando se la requería [8].

En las últimas décadas se han creado entornos visuales de programación, tales como Scratch [10], Pilas Bloques [19] o Alice [11] que eximen a los usuarios de la responsabilidad de dominar la sintaxis del lenguaje de programación, pudiendo centrarse en la semántica del programa [9]. Por otro lado, es importante considerar que a lo largo de la historia, diferentes tecnologías han posibilitado múltiples aprendizajes ya que la mediatización en el devenir histórico es una característica universal de todas las sociedades humanas [14]. En este sentido, las herramientas no son neutrales, no se configuran como un fin en sí mismo ni son elementos pasivos cuando son utilizadas en una situación, sino que van a co-construir la actividad de la que forman parte, posibilitando un aprendizaje situado [12] entendido como práctica social [13]. Sin embargo, aunque lo que sabemos y qué tipo de problemas podemos resolver está ampliamente determinado por los recursos y las herramientas que tengamos a nuestro alcance en una práctica social específica, predomina un uso instintivo y no reflexivo de las herramientas usadas, que se vuelven invisibles para nosotros [15].

Por otro lado, Brennan y Resnick [17] plantean que conocer la definición de un concepto no es útil si no está acompañada por la familiaridad y la fluidez en las prácticas, que permiten poner los conceptos en acción dentro de una creación en particular.

En este contexto y en el marco de experiencias innovadoras y pioneras de introducción de las CC en el contexto argentino, resulta necesario conocer las experiencias de los docentes cuando se acercan a la programación: ¿Qué características presentan los programas

---

<sup>1</sup> Para más información visitar: <http://program.ar/especializaciones-de-nivel-superior-en-ensenanza-de-las-ciencias-de-la-computacion/>

diseñados? ¿Qué problemas encuentran al realizar sus primeros programas? ¿Pueden poner en práctica los conceptos aprendidos? ¿Cómo encuentran las soluciones?

Considerando estas cuestiones, este trabajo presenta un análisis preliminar de producciones elaboradas en el marco de un trayecto de formación docente del cual participaron maestros del nivel primario. Con ello se pretende proponer directrices que permitan informar futuros diseños curriculares en el campo de la enseñanza de las Ciencias de la Computación y particularmente de la programación.

## 2 Marco metodológico

Este estudio exploratorio y descriptivo se desarrolló en el marco de un proyecto de investigación científica y tecnológica (PICT)<sup>2</sup> que toma como caso de estudio a una especialización docente de nivel superior en Didáctica de las Ciencias de la Computación, desarrollada en la ciudad de Rosario (convenio UNR, Fundación Sadosky, Ministerio de Educación de la Pcia. de Santa Fe) [18]. Esta especialización constituye una experiencia piloto gratuita destinada a docentes de nivel primario [16], de dos años de duración. Particularmente, este estudio se centra en el módulo “Taller de Programación 1” perteneciente al primer cuatrimestre y que se dictó en paralelo con los módulos “Introducción a las Ciencias de la Computación” y “Pensamiento Computacional 1”.

### 4.2 Contexto y participantes

El taller de programación comprendió 10 clases de tres horas y dos de una hora y media, que tuvieron lugar cada dos semanas. Su objetivo era introducir estructuras algorítmicas básicas, el uso de secuencias, condicionales, repeticiones, captura de eventos, variables y manejo de mensajes.

Para lograr esto, se usaron diferentes entornos de programación visual y de bloques, comenzando con plataformas que ofrecen pequeños desafíos de programación, como Light-Bot y Pilas Bloques, para luego presentar Scratch que permite programar juegos, animaciones, historias y diferentes tipos de artefactos multimedia. Además, se emplearon diferentes materiales y recursos, especialmente diseñados para programadores novatos, tales como las guías de Scratch, las cartas de Scratch y los manuales de Program.ar<sup>3</sup> que permitieron acompañar el proceso de introducción de los conceptos de programación, ya que un 74% de los 43 docentes de primaria participantes no había programado antes.

Asimismo, del total de participantes 38 eran mujeres y ocho hombres, el 80% de ellos tenía entre 31 a 50 años. La mayoría de los maestros, un 60%, tenía entre siete y 20 años de experiencia docente y el 76% había tomado cursos sobre TIC y educación. También participaron del módulo cuatro profesores que coordinaron el taller, tres con formación en

---

<sup>2</sup> Trabajo financiado por ANPCyT-FonCyT PICT 2016-1530.

<sup>3</sup> Estos recursos se encuentran disponibles en los siguientes sitios web: <http://program.ar/manual-primer-ciclo-primaria/>; <https://scratch.mit.edu/ideas>; <http://pilasbloques.program.ar/>

Ciencias de la Computación y uno con formación en pedagogía y experiencia en formación docente en programación.

#### 4.2 Recolección y análisis de datos

Durante el taller se recogieron diferentes tipos de datos en distintos formatos. Para este estudio en particular, los datos están conformados por 17 programas realizados con Scratch como trabajos finales del módulo y 17 informes escritos que se les solicitó entregar junto con sus proyectos.

La consigna para la realización del proyecto final planteaba algunos requisitos vinculados con los conceptos trabajados a lo largo del módulo. En este sentido se solicitaba que los proyectos tuvieran:

- al menos 3 objetos diferentes (uno de ellos que tenga al menos dos disfraces que cambien en algún momento) y 2 escenarios (que cambien en diferentes momentos)
- incluir al menos tres tipos de bloques de eventos diferentes
- secuencias
- condiciones
- repeticiones
- variables
- mensajes
- el tema del proyecto fue libre pero la producción debía tener una trama clara y coherente.

Por su parte, el informe, como muestra la tabla 1, planteaba preguntas reflexivas que apuntaban a relevar información sobre diferentes dimensiones relacionadas con el proceso de creación y programación<sup>4</sup>. Tanto los proyectos como los informes fueron realizados en grupos de 2 o 3 participantes.

**Tabla 1.** Dimensiones exploradas en el informe final.

Dimensión	Consignas y preguntas asociadas
Descripción del proyecto	Describir brevemente el proyecto (Por ej; si es un juego cuáles son las reglas y cómo se juega). Especificar en qué versión de Scratch se trabajó (recuerden que lo que se realiza en Scratch 1.4 puede abrirse con Scratch 2.0 pero no a la inversa).
Descripción del proceso de creación	- ¿Cómo lo crearon?, ¿qué diferentes cosas intentaron?, ¿qué revisiones realizaron al proyecto original? - ¿Cómo decidieron qué objetos utilizar y dónde debían ir?

<sup>4</sup> Para esta instancia se tomaron como referencias las propuestas para la evaluación disponibles en el siguiente enlace: <http://scratched.gse.harvard.edu/ct/assessing.html>

	¿Cómo eligieron los programas y su organización dentro del proyecto? - Si se inspiraron en algún otro proyecto, mencionar qué ideas tomaron y para qué las usaron.
Descripción de dificultades	Descripción de un momento en que algo no haya funcionado como esperaban: ¿Qué ocurrió? ¿Cómo investigaron el problema? ¿Pudieron solucionarlo? ¿Cómo?
Descripción de mejoras y futuras revisiones	¿Cómo podrían mejorar su trabajo? ¿Hay puntos que quisieron desarrollar y no llegaron? ¿Por qué? ¿Dificultades? ¿Comentarios finales?

Los datos obtenidos fueron sometidos a un análisis estadístico descriptivo así como a un análisis de contenido, lo que permitió obtener frecuencias, medias y demás datos generales que luego se complementaron con ejemplos y citas que permiten complejizar el análisis y la comprensión del proceso, a partir de la construcción de categorías inductivas y codificaciones sucesivas. En estos procesos se contó con la ayuda de Atlas Ti 7 y SPSS 20.

## 4 Resultados y discusión

En esta sección se presenta una síntesis de los resultados obtenidos, atendiendo a las limitaciones propias del escrito. En un primer momento se reseñan características generales de los proyectos para luego describir cómo se presentaron en los mismos los conceptos de programación trabajados, y de qué manera se resolvieron las situaciones problemáticas.

### 4.2 Características de los proyectos

Los proyectos realizados grupalmente empleando la herramienta Scratch cuentan con algunas características particulares.

Como refleja la tabla 2, aunque en las clases de la especialización se utilizó mayormente la versión de Scratch 1.4 debido a las posibilidades técnicas de las netbooks utilizadas, el 59% de los proyectos se realizaron en la versión 2.0, que presenta algunas diferencias con respecto a su versión anterior.

**Tabla 2.** Versiones de Scratch empleadas en los proyectos

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Scratch 1.4	7	41,2	41,2	41,2
	Scratch 2	10	58,8	58,8	100,0

Total	17	100,0	100,0
-------	----	-------	-------

Estas diferencias refieren particularmente a la disposición de los diferentes componentes de la interfaz y a la incorporación de nuevas categorías de bloques. Asimismo, la versión 2.0 permite el trabajo en línea a diferencia de la versión 1.4 que solo habilita al trabajo offline. Estas posibilidades y diferencias fueron presentadas en las clases del taller y podría decirse que no han constituido un obstáculo para la realización de los proyectos en las diferentes versiones. Este dato permite pensar que las diferencias no resultan significativas y que la gran mayoría de los participantes pudo trabajar de manera indistinta en ambas, lo cual resulta en una fortaleza a la hora de elegir esta herramienta, ya que generalmente los cambios en las interfaces pueden ser motivos de frustración y/o abandono de las actividades por parte de los docentes que se inician en estas temáticas.

Por otro lado, como se observa en la tabla 3, el 65% de los proyectos fueron juegos, mientras que el 35% restante se corresponde con historias o animaciones. Los juegos permiten interacciones, proponiendo al usuario acciones a realizar tales como apretar una tecla, introducir una respuesta o mover a un personaje. Las animaciones o historias plantean más bien un rol pasivo al usuario, limitado a observar lo que ocurre sin posibilidad de interacción.

**Tabla 3.** Tipos de producciones

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Juego	11	64,7	64,7	64,7
	Animación	6	35,3	35,3	100,0
	Total	17	100,0	100,0	

En cuanto al contenido sobre el que versan los proyectos, se realizó una clasificación de los mismos considerando si la temática refería a contenidos propios del nivel primario, a contenidos más bien transversales o a temáticas recreacionales, ligadas más bien al juego y a la diversión sin plantear una finalidad educativa.

Vale señalar que la consigna del trabajo final no requería que el contenido del proyecto sea educativo, pero como muestra la tabla 4, la gran mayoría de los participantes optó por crear un proyecto vinculado a la educación, ya sea en relación a contenidos curriculares o transversales.

En este sentido, el 41% de los proyectos trabajaron contenidos curriculares y el 30% transversales, mientras que el 30% restante corresponde a proyectos recreacionales. Por lo tanto, un 70% de los proyectos está relacionado con contenidos que pueden trabajarse en la escuela y que resultan de interés para los docentes participantes.

**Tabla 4.** Categorías temáticas de las producciones

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Contenido curricular	7	41,2	41,2	41,2
	Temática transversal	5	29,4	29,4	70,6
	Recreacional	5	29,4	29,4	100,0
	Total	17	100,0	100,0	

A continuación, la tabla 5 presenta un ejemplo de los tipos de proyectos presentados que permite observar las diferencias entre las categorías señaladas.

**Tabla 5.** Ejemplos de tipos de proyectos.

Contenido	Tipo de proyecto	Proyecto ejemplo
Recreacional	Juego	P1: juego de atrapar, donde el gato de Scratch tiene que atrapar murciélagos para salvar a una la abuela que está en su habitación, evitando que lo toque un elefante volador que le resta puntos.
	Animación	P5: una rana se mueve y atrapa a una mariposa que está parada. Cuando esto sucede, aparece un signo de "Has ganado" y el fondo cambia.
Curricular	Juego	P14: juego sobre alfabetización donde el jugador tiene que reorganizar las letras de un nombre siguiendo instrucciones orales de un asistente o una bruja.
	Animación	P10: cuento de hadas narrado y animado con un canguro, un pingüino y un zorro que brindan un mensaje sobre la humildad.



	Juego	P3: juego sobre hábitos saludables donde un niño tiene que tomar 8 vasos de agua mientras camina por un laberinto y evita a otro personaje que se mueve al azar a través de su camino.	
Transversal	Animación	P15: animación sobre convivencia escolar protagonizada por dos niños que discuten por un malentendido en la escuela y una niña que plantea una reflexión sobre lo que ha sucedido.	

Estos ejemplos ponen de relieve diferentes maneras de crear, producir y generar materiales educativos, mientras se aprende a programar con una herramienta concreta.

#### 4.2 Conceptos de programación

Retomando la consigna del trabajo final, donde se planteaban ciertos requerimientos de los proyectos, es posible observar que la gran mayoría de los grupos pudo responder a los mismos. En este sentido, los resultados mostraron que la mayoría de los programas realizados por los participantes ponen en acción los conceptos trabajados en clases, observándose en general, un menor uso de repeticiones y variables, tal como muestra la figura 1.

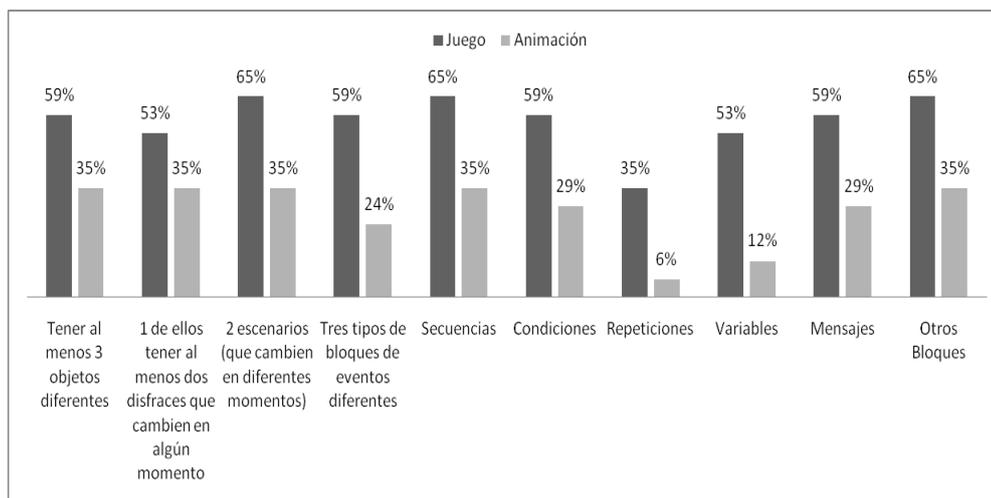


Fig. 1. Características de los proyectos finales programados.

Asimismo, cabe señalar que como se observa en el gráfico, se relevaron diferencias en relación con el tipo de producción. En los proyectos clasificados como animaciones se relevó menor uso de eventos, objetos, variables, condiciones, repeticiones y mensajes que en los proyectos que plantearon juegos. Asimismo, se observaron diferencias en la cantidad de objetos y escenarios utilizados en los dos tipos de producciones, como reflejan las figuras 2 y 3.

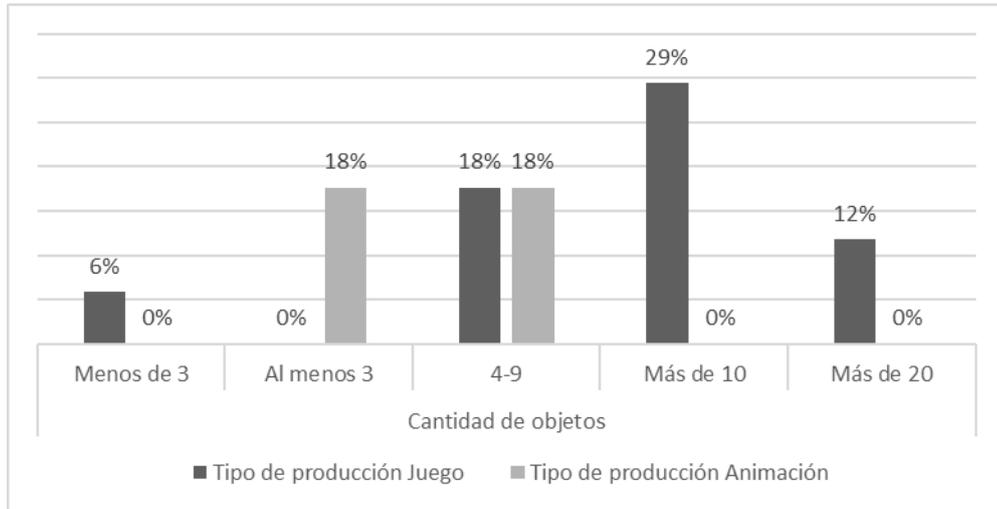


Fig. 2. Cantidad de objetos presentes en los proyectos según tipo de producción.

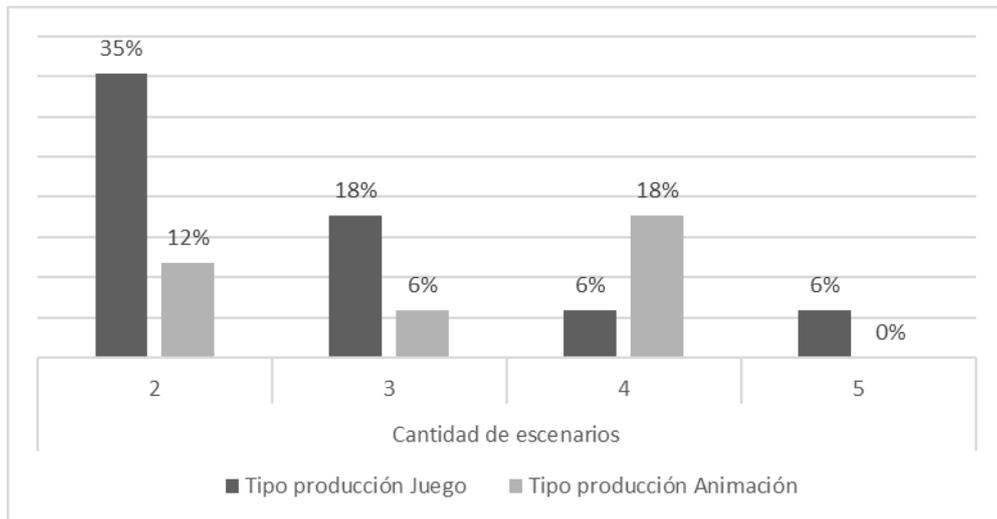


Fig. 3. Cantidad de escenarios en los proyectos según tipo de producción.

En los juegos se observa mayor presencia de objetos y escenarios que en las animaciones. Observando las figuras, un 29% de los juegos contaron con más de 10 objetos y un 12% con más de 20, mientras que la mayoría de las animaciones presentaron entre 4 y 9 objetos. Por su parte, el 35% de los juegos presentó 2 escenarios, aunque el 18% de las animaciones contó con más de 4. En este sentido, pareciera que los juegos requirieron más cantidad de objetos pero no necesariamente de escenarios.

## 4.2 Resolución de situaciones problemáticas

Por otra parte, el análisis de los informes, particularmente en torno a la dimensión ligada a las dificultades, permitió identificar diferentes situaciones problemáticas que los participantes señalaron durante el proceso de programación. En ellas se observa tanto un desconocimiento inicial del funcionamiento de algunos bloques como dificultades para aplicar ciertos conceptos de programación.

Sin embargo, además de las situaciones problemáticas, los participantes señalaron las soluciones que emplearon y los recursos utilizados para ello, lo cual les permitió resolver estas dificultades encontradas durante el proceso de creación y programación.

Dentro de los recursos mencionados por los participantes, se observa un 40% de instancias que remiten a procesos de revisión y depuración de programas, seguido de un 25% que destaca el andamiaje docente. En menor medida, se destaca la búsqueda de información online (9%), el uso de las guías de Scratch o las fichas de ejercicio (16%) y la exploración de bloques disponibles (6%). En algunos casos se señalan varios recursos empleados para la resolución de una misma problemática. La tabla 6 muestra algunos ejemplos de estas categorías extraídas de los informes escritos, donde en palabras de los propios docentes las diferentes maneras de resolución.

**Tabla 6.** Ejemplos de recursos empleados y formas de resolución.

Tipo de recurso	Ejemplos de extractos discursivos
Revisión y depuración de programas	<p>“Probando diferentes programas pudimos hacer que los animales y demás objetos quedasen ocultos y aparezcan cuando iniciara el juego.”</p> <p>“Recorrimos los controles para identificar las fallas y notamos que habíamos programado que las langostas al ser tocadas por el pez esperen 1 segundo.”</p>
Andamiaje docente	<p>“El programar junto a los profesores en clase facilitó el camino en muchos aspectos. Por ejemplo, en lugar de escribir reiteradas veces lo mismo en diferentes cosas, copiamos la programación de un objeto a otro, arrastrando el bloque al objeto deseado.”</p> <p>“En esta ocasión consultamos al profesor que nos explicó cómo trabajar con los bloques de operadores matemáticos</p>

	de igualdad.”
Uso de guía de Scratch o fichas de ejercicio	“Pensamos y probamos distintos movimientos, hasta que guiándonos por las cartas de Scratch que teníamos encontramos la instrucción de movimiento "fijar estilo de rotación" y usando un control de repitencia pudimos lograr que el canguro salte.” “Probamos veinte veces y recurrimos a la carta n° 7 "Bailar Twist" que nos recordó lo trabajado en clase. Así lo modificamos y logramos incorporar el sonido a cada cambio de escenario.”
Búsqueda de información online	“Investigamos en internet varias maneras de hacerlo.” “Para solucionarlo buscamos ejemplos en la página de Scratch de otros juegos que utilizaban esta dinámica.”

Se observa que el andamiaje docente así como la revisión de programas implican un trabajo en colaboración con otros, que aparece como una opción destacada al momento de resolver las situaciones problemáticas. Asimismo, los materiales ofrecidos por la comunidad de Scratch y particularmente por la comunidad docente de Scratch (ScratchEd) también aparecen como recursos válidos que guiaron los procesos de resolución, permitiendo a los participantes encontrar soluciones a sus situaciones problemáticas, adaptando las propuestas de solución a la particularidad del caso. En este sentido, emplear una herramienta ampliamente utilizada por la comunidad internacional que dispone de múltiples materiales específicos, parece ser una opción acertada al pensar recursos propicios para un curso introductorio sobre programación.

#### 4 Conclusiones

Este análisis permite inferir diferentes cuestiones relevantes para repensar la didáctica de la programación, especialmente en el contexto de la formación de docentes en nivel primario.

En cuanto a las características de los proyectos, la mayoría se realizó en la versión 2.0 de Scratch, encontrándose una mayor cantidad de juegos que de animaciones y de proyectos con contenidos educativos en vez de meramente recreativos. Asimismo, una comparativa entre juegos y animaciones reveló que en los juegos se despliegan mayor cantidad de bloques vinculados a distintos conceptos de programación y se observa una mayor cantidad de objetos que en las animaciones, donde sí se emplearon más escenarios. Para la resolución de situaciones problemáticas los participantes recurrieron en gran medida al trabajo colaborativo con docentes y pares y señalaron el empleo de múltiples recursos propios de la comunidad de Scratch.

En relación al diseño de propuestas de formación docente en estas temáticas, se observa que Scratch aparece como una herramienta apropiada para introducir a los docentes de primaria en el mundo de la programación, permitiéndoles la creación de producciones vinculadas con su práctica docente y con contenidos educativos. La posibilidad de crear

proyectos vinculados a diversas temáticas y contenidos, hace que sea posible de ser empleada por docentes interesados en diferentes áreas curriculares así como también en problemáticas transversales propias de cada comunidad o contexto, donde cobra aún mayor relevancia la opción de creación personalizada.

Por otra parte, la creación de proyectos del tipo juegos surge como una opción que permite un mayor despliegue de bloques y por lo tanto, programas más complejos. En este sentido, se recomienda su uso en lugar de animaciones o historias, si lo que se espera es un acercamiento a conceptos de programación más avanzados.

Finalmente, la programación en equipos o grupos de trabajo parece ser una estrategia didáctica apropiada para los momentos iniciales de formación, así como la presencia de docentes especialistas en el tema que permitan resolver dudas in situ. Por ende, las clases presenciales aparecen como una propuesta indicada para un primer abordaje de la temática.

## Referencias

1. S. Grover, R. Pea, and S. Cooper, "Designing for deeper learning in a blended computer science course for middle school students," *Comput. Sci. Educ.*, vol. 25, no. 2, pp. 199–237, 2015.
2. F. Heintz, L. Mannila, K. Nygård, P. Parnes, and B. Regnell, "Computing at School in Sweden – Experiences from Introducing Computer Science within Existing Subjects," in *Informatics in Schools. Curricula, Competences, and Competitions*, vol. 9378, A. Brodnik and J. Vahrenhold, Eds. Springer, 2015, pp. 69–81.
3. A. Yadav, "Computer Science Teacher Professional Development: Towards a Research Agenda on Teacher Thinking and Learning," in *WiPSCE '17*, 2017, pp. 1–2.
4. T. E. Reding and B. Dorn, "Understanding the 'Teacher Experience' in Primary and Secondary CS Professional Development," in *Proceedings of the 2017 ACM Conference on International Computing Education Research*, 2017, pp. 155–163.
5. C. Brackmann, D. Barone, A. Casali, R. Boucinha, and S. Muñoz-Hernandez, "Computational thinking: Panorama of the Americas," in *2016 International Symposium on Computers in Education, SIIE 2016: Learning Analytics Technologies*, 2016, pp. 1–6.
6. N. Monjelat and P. San Martín, "Programar con Scratch en contextos educativos: ¿Asimilar directrices o co-construir Tecnologías para la Inclusión Social?," *Prax. Educ.*, vol. 20, no. 1, pp. 61–71, 2016.
7. S. Papert, *Mindstorms: Children, computers and powerful ideas*. New York: Basic Books, 1980.
8. K. Brennan, "Best of both worlds: issues of structure and agency in computational creation, in and out of the school," *Massachusetts Institute of Technology*, 2013.
9. M. Moallem, S. P. Morge, S. Narayan, and G. A. Tagliarini, "The Power of Computational Modeling and Simulation for Learning STEM Content in Middle and High Schools," in *Improving K-12 STEM Education Outcomes through Technological Integration*, D. Falvo and M. Urban, Eds. Hershey PA, USA: IGI Global, 2016.
10. M. Resnick et al., "Scratch: Programming for All," *Commun. ACM*, vol. 52, no. 11, p. 60, 2009.
11. W. Dann, S. Cooper, and R. Pausch, *R. Learning to Program with Alice, Second Edition*. Pearson, 2009.
12. J. Lave and E. Wenger, "Legitimate Peripheral Participation in Communities of Practice," in *Situated learning: legitimate peripheral participation*, Cambridge, UK: Cambridge University Press, 1991.
13. L. S. Vygotsky, *Mind in society: The development of higher psychological processes*. London, UK: Harvard University Press, 1978.
14. E. Verón, *La semiosis social 2. Ideas, momentos, interpretantes*. Buenos Aires: Paidós, 2013.

15. J. Wertsch, *Voices of the mind. A sociocultural approach to Mediated Action*. Cambridge, Mass: Harvard University Press, 1991.
16. A. Casali, D. Zanarini, N. Monjelat, and P. San Martín, "Teaching and Learning Computer Science for Primary School Teachers: an Argentine Experience," in *Proceedings of LACLO 2018*, 2018, pp. 1–8.
17. K. Brennan and M. Resnick, "New frameworks for studying and assessing the development of computational thinking," *Annu. Am. Educ. Res. Assoc. Meet. Vancouver, BC, Canada*, pp. 1–25, 2012.
18. A. Casali, D. Zanarini, P. S. San Martín, and N. Monjelat, "Pensamiento Computacional y Programación en la Formación de Docentes del Nivel Primario," in *XX Workshop de Investigadores en Ciencias de la Computación*, 2018, pp. 451–455.
19. A. Sanzo, F. Schapachnik, P. Factorovich and F. S. O'Connor, "Pilas bloques: A scenario-based children learning platform," 2017 Twelfth Latin American Conference on Learning Technologies (LACLO), La Plata, 2017, pp. 1-6. doi: 10.1109/LACLO.2017.8120926