

El Pensamiento Computacional en la Modelización de la Ingeniería Genética

Gimena B. Fussero¹, Maricel Occelli² y Marcela Chiarani³

¹Becaria SECyT. Grupo EDUCEVA.CienciaTIC, Departamento de Enseñanza de la Ciencia y la Tecnología, Facultad de Ciencias Exactas, Físicas y Naturales, Universidad Nacional de Córdoba, Córdoba, Argentina.

²Grupo EDUCEVA.CienciaTIC, Departamento de Enseñanza de la Ciencia y la Tecnología, Facultad de Ciencias Exactas, Físicas y Naturales, Universidad Nacional de Córdoba, CONICET, Córdoba, Argentina.

³Departamento de Informática, Facultad de Ciencias Físico, Matemáticas y Naturales, Universidad Nacional de San Luis, San Luis, Argentina.

Fecha de recepción del manuscrito: 22/03/2019

Fecha de aceptación del manuscrito: 06/02/2020

Fecha de publicación: 31/10/2020

Resumen—El Pensamiento Computacional, hace un tiempo, se está incorporando a los espacios educativos. Sin embargo, no se conoce mucho se su inclusión en la Enseñanza de la Biología. Es por ello que se buscó incorporar dicho pensamiento, a través del lenguaje de la programación Scratch, a la enseñanza de la Ingeniería Genética. Para ello se trabajó en un curso de 5° año (N=34) de una escuela secundaria de la ciudad de Córdoba durante 6 semanas consecutivas. Durante dicho periodo se desarrolló una secuencia didáctica y se recolectaron datos que luego fueron analizados siguiendo una metodología cualitativa realizando una triangulación metodológica. Los principales resultados muestran que la utilización de Scratch favorece el desarrollo de algunas habilidades del Pensamiento Computacional a la vez que los estudiantes modelizan para aprender sobre Ingeniería Genética.

Palabras clave—Pensamiento computacional, Scratch, Modelización, Ingeniería Genética, Escuela secundaria.

Abstract—Write in English the same text written above.

Keywords— Computational thinking, Scratch, Modeling, Genetic Engineering, High school.

INTRODUCCIÓN

Los objetos programables controlados por software se encuentran en todos los aspectos de la vida cotidiana creando ecosistemas digitales (Manovich, 2013). En este escenario tener la habilidad de manejar el lenguaje de las computadoras se considera un nuevo alfabetismo indispensable para participar de la realidad digital (Román-González, 2015), se trata de, según Rushkoff (2010) de programar o ser programado. Sumado a lo anterior y dado el creciente rol que ocupa la informática en la educación, la comprensión de cómo las personas interactúan con la computación y aprenden a pensar a través de lenguajes informáticos se convirtió en un área de interés para las investigaciones en educación y medios de comunicación (National Research Council, 2010).

En este sentido, existe una tendencia para incorporar el Pensamiento Computacional (PC) a la educación obligatoria para contribuir a la formación de estudiantes con habilidades para ser participantes activos y reflexivos en la cultura

digital (Gretter y Yaday, 2016). Sin embargo, a pesar que el concepto de PC fue propuesto en el año 2016 por Jeanette Wing (Seymour Papert realizó una primera aproximación del mismo en el año 1980), todavía no existe un acuerdo en la comunidad científico-educativa respecto a su definición (Gouws *et al.*, 2013). En un principio implicaba la resolución de problemas, el diseño de sistemas y la comprensión de la conducta humana haciendo uso de los conceptos fundamentales de la informática (Wing, 2006). Posteriormente, en el año 2008, Wing realizó una revisión de la definición original enunciando que “El pensamiento computacional incluye los procesos de pensamiento implicados en la formulación de problemas y de sus soluciones de tal modo que queden representados de una manera que pueda ser abordada de forma efectiva por un agente-procesador de la información, como una computadora. Al año siguiente, la Sociedad Internacional para la Tecnología en la Educación (*International Society for Technology in Education*, ISTE) junto con la Asociación de Docentes en Ciencias de la Computación (*Computer Science Teachers Association*, CSTA) y demás colaboradores desarrollaron una definición operacional del PC. Así el mismo quedó definido como “Un proceso de resolución de problemas (que incluye, pero no se limita a) las siguientes características: *Formulación de problemas de una manera

Dirección de contacto:

Gimena Fussero, Avenida Vélez Sarsfield 299, X5022 QXF. Tel: 4332113 interno 23, gimfussero@gmail.com

que permite usar una computadora y otras herramientas para ayudar a resolverlos, *Organizar y analizar lógicamente datos, *Representación de los datos a través de abstracciones como modelos y simulaciones, *Automatización de soluciones a través del pensamiento algorítmico (serie de pasos a seguir), *Identificación, análisis e implementación de posibles soluciones con el objetivo de lograr la combinación más eficiente y efectiva de pasos y recursos y *Generalización, es decir transferir este proceso de resolución de problemas a una amplia variedad de problemas.

Destacan que el PC es un enfoque para la resolución de problemas que permite la integración de las tecnologías digitales con las ideas humanas no reemplazando la creatividad y el pensamiento crítico sino que reforzando dichas habilidades (ISTE y CSTA, 2011). Ulteriormente, en el año 2011, para aportar a la discusión, Wing propuso una nueva definición de PC: “El pensamiento computacional es el proceso de pensamiento involucrado en la formulación de problemas y sus soluciones para que las mismas se representen de una manera que puedan ser efectivamente ejecutadas por un agente de procesamiento de la información” (pudiendo ser un humano, una máquina o más frecuentemente una combinación entre ambos). De dicha definición se desprenden dos aspectos significativos para la educación. Por un lado el PC es un proceso de pensamiento, por lo tanto es independiente de la tecnología y por otro lado el PC es un tipo específico de resolución de problemas (Bocconi *et al.*, 2016). En síntesis, el PC implica pensar como un científico de la computación al enfrentarse a un problema (Román-González, 2015) pero no se limita a ello. Lye y Koh (2014) afirman que se trata de una alfabetización digital para crear, remezclar y compartir recursos digitales posibilitando no sólo consumir tecnología sino crearla. En este sentido, el PC involucra una serie de habilidades que ofrecen diferentes potencialidades al ser introducido en la enseñanza obligatoria. Lee y colaboradores (2011) indican que el PC puede permitir que niños y jóvenes piensen de manera diferente al enfrentarse a un problema desde una perspectiva diferente.

Una de las maneras de caracterizar al PC es mediante las siguientes habilidades: Paralelismo, Pensamiento lógico, Control de flujo, Interactividad con el usuario, Representación de la información, Abstracción y Sincronización (Moreno León *et al.*, 2015). De dichos conceptos, la abstracción ocupa el nivel más alto dentro del PC ya que implica decidir que detalles se deben resaltar y cuales se pueden ignorar de modo que el problema sea más fácil de comprender sin perder información importante logrando una buena representación del sistema (Wing, 2008). Se utiliza para permitir que un objeto represente a muchos y para capturar las propiedades esenciales comunes a un conjunto de objetos mientras que la información no relevante se oculta (Wing, 2011). En este sentido, se considera a la abstracción como la esencia del PC siendo las herramientas “mentales” de la computación (Wing, 2008).

Dentro de la informática, la programación conforma una de sus áreas. La misma no debe ser entendida como un sinónimo de codificación ya que implica una actividad más amplia que involucra el análisis de problemas, el diseño y la implementación de su solución estando la codificación en la etapa de implementación a través de un lenguaje de

programación particular (Bocconi *et al.*, 2016). Los diferentes lenguajes de programación contribuyen al desarrollo de las llamadas “competencias del siglo XXI” destacándose entre ellas el pensamiento crítico, el manejo de la información, la comunicación efectiva y el trabajo colaborativo (Sunkel y Trucco, 2012). De hecho, la programación hace tiempo se ha incorporado a la educación (Papert en el año 1967 introdujo LOGO) y en los últimos años su adhesión fue aumentando. Los motivos son de diversa índole. Con la programación se promueve el desarrollo de competencias claves al posicionar a los estudiantes frente a situaciones problemáticas que necesitan de un trabajo en equipo para su resolución (Vázquez-Cano y Ferrer Delgado, 2015). En este sentido la programación como competencia cognitiva permite el aprendizaje social estableciendo nuevas ecologías de aprendizaje (Valdeverde Berrocoso *et al.*, 2015).

Al respecto, Scratch constituye una de las más grandes comunidades de aprendizaje siendo utilizado masivamente en todo el mundo con más de 7 millones de proyectos compartidos por niños, niñas y jóvenes (Moreno-León *et al.*, 2015). Este lenguaje de programación fue diseñado en el Lifelong Kindergarten del Media Laboratory del MIT (Universidad de California, Los Ángeles) por Michael Resnick y colaboradores. Uno de los propósitos principales es que la programación con Scratch se utilice como herramienta para desarrollar habilidades más allá de la programación y pueda apoyar y potenciar el aprendizaje en otras disciplinas (Resnick, 2013). La programación en Scratch se basa en un conjunto de instrucciones ícono-textuales que se combinan para crear programas. Dichas instrucciones se encuentran en forma de bloques que sólo se acoplan si la sintaxis es correcta lo que libera al usuario de la complejidad de escribir códigos y le permite concentrarse en la solución del problema. Además ayuda al aprendizaje mediante el PC (Brennan y Resnick, 2012) cuando situaciones y desafíos son planteados desde afuera como una actividad problematizadora para los estudiantes (Astudillo *et al.*, 2016). Dadas estas características, y otras, Scratch se está utilizando en todos los niveles de educación formal, desde colegios a universidades (Moreno-León y Robles, 2015; Malan y Leitner, 2007). Además permite, a través de la abstracción, que los estudiantes generen modelos para explicar fenómenos, expresarlos y revisarlos, es decir, desarrollen la práctica científica de la modelización.

En la didáctica de las ciencias, en los últimos años, estas prácticas comenzaron a considerarse como una competencia a desarrollar dentro de la educación científica. Particularmente, la modelización adquirió un notorio interés dentro de la didáctica (Adúriz-Bravo e Izquierdo, 2009) ya que al involucrar a los estudiantes con esta práctica los ayuda a comprender las ideas fundamentales de las distintas disciplinas científicas como así también su conocimiento epistemológico (Lehrer y Schauble, 2006). En este contexto, la modelización es entendida como el proceso de aprendizaje que acompaña al trabajo con modelos (Justi y Gilbert, 2002) entendidos estos como la representación de un objeto, fenómeno o sistema con el propósito de describir, explicar o predecir el comportamiento de una parte del mundo real que se quiere modelizar (Adúriz-Bravo, 2012). En concordancia con lo anterior, en la enseñanza de las ciencias los modelos ocupan un papel central dado que contribuyen a que los

estudiantes comprendan los modelos consensuados por la comunidad científica permitiéndoles comprender algún fenómeno o sistema natural construyendo un modelo de los mismos, actuando estos como un puente entre el conocimiento científico y el fenómeno (mundo) real (Gilbert, 2002). En este sentido, se espera que los estudiantes, cuando construyen sus modelos sean capaces de pensar en explicaciones multicausales, de evaluar diferentes sistemas, de inferir a partir de datos, entre otras actividades características de las tareas científicas y, por tal motivo, el foco debe estar puesto tanto en el proceso como en los productos de modelización obtenidos (Gómez Galindo, 2014). Tal proceso de modelización incluye desde la generación de los modelos mentales hasta la expresión y revisión de los mismos (Justi y Gilbert, 2002) constituyéndose como una actividad epistémica que demanda una diversidad de capacidades tanto de índole cognitivo como metacognitivo (Oliva y Aragón, 2009; Torres y Vasconcelos, 2017). Por consiguiente, la práctica científica de la modelización no sólo incluiría elaborar los modelos sino también revisarlos entendiendo su utilidad, sus limitaciones y su carácter aproximativo y cambiante (Aragón *et al.*, 2018).

La construcción de modelos es esencial para la comprensión de conceptos relacionados con la genética y el caso particular de la Ingeniería Genética (IG) no constituye una excepción. El nivel de abstracción de los conceptos de genética requieren para su aprendizaje de la construcción de representaciones mentales (Diez de Tancredi y Caballero, 2004), es decir, de modelos que les permitan transformar ideas abstractas en teorías inteligibles (Venville y Donovan, 2008). Dichos modelos contribuirán al desarrollo de una alfabetización científica dado que los conceptos genéticos están cada vez más presentes en la vida cotidiana llegando desde diferentes fuentes (Caballero, 2008) y requiriendo de la toma de decisiones de naturaleza local por parte de los ciudadanos y los mismos deben estar preparados para hacerlo (Venville y Donovan, 2007). En estos momentos, varios de los aspectos de la vida cotidiana están relacionados directa o indirectamente con los procesos biotecnológicos (de la Vega Naranjo *et al.*, 2018). En este sentido, la construcción de conceptos como ADN, cromosomas, genes y código genético resultan elementales para la comprensión de avances científicos y sus implicaciones en el día a día de las personas (Pedrancini *et al.*, 2007).

En el ámbito educativo y en lo que respecta a la IG en particular, a pesar que lleva varios años incluida en los planes de estudio de Biología en Argentina a través de varias reformas educativas, investigaciones muestran debilidades conceptuales por parte de los estudiantes (Ocelli *et al.*, 2011). En esta línea, Lewis y Wood-Robinson (2000) observaron una falta de conocimientos respecto a los principios básicos de la IG. Por su parte Pedrancini y colaboradores (2008) encontraron que el significado de “genéticamente modificado” es fragmentado y confuso si bien los estudiantes se encuentran familiarizados con dicha terminología.

En cuanto a las actitudes de los estudiantes frente a procesos y productos derivados de la IG se encontró que las personas mejor formadas al respecto tenían opiniones más favorables (Tegegne *et al.*, 2013). Sin embargo, la enseñanza

tal cual está siendo organizada no favorece el desarrollo conceptual de dicho campo de manera que los conocimientos adquiridos en la escuela no siempre posibilitan superar el sentido común (Pedrancini *et al.*, 2007, 2008). En este escenario, los docentes pueden utilizar estos desarrollos científico-tecnológicos para tratar las implicancias de IG evitando las tendencias reduccionistas y determinista y abordando una enseñanza de forma transdisciplinar y contextualizada promoviendo una enseñanza que permita a los ciudadanos una construcción de conocimientos sobre la cual tomas decisiones reflexivas y críticas (Roa Acosta y Valbuena Ussa, 2013).

Entonces, teniendo en cuenta lo anteriormente expuesto, se plantea como objetivo de este trabajo analizar cuál es el grado de desarrollo de las habilidades del PC cuando estudiantes construyen modelos de IG utilizando el lenguaje de programación Scratch.

METODOLOGÍA

En esta investigación se utilizó la metodología conocida como *Estudios o Experimentos de Diseño* los cuales se orientan a generar conocimientos que ayuden a mejorar las prácticas de enseñanza (Rinaudo y Donolo, 2010). En este contexto, la expresión diseño hace referencia al diseño instructivo que se prepara, implementa y analiza (Confrey, 2006) con el propósito de incidir en los aprendizajes produciendo modificaciones tendientes a mejorarlos (Rinaudo y Donolo, 2010).

La investigación se llevó a cabo en una escuela secundaria de la ciudad de Córdoba (Argentina) en el espacio curricular de Biología en un 5° año la cual tiene prescripta IG para la especialidad en Ciencias Naturales. Participaron 34 estudiantes, de ambos géneros, de entre 16 y 17 años. La organización del estudio constó de tres fases: preparación, implementación y análisis.

En este trabajo se presenta parte de la fase de análisis del diseño, en particular lo referente a las habilidades del PC. Para dicho análisis se utilizó Dr. Scratch (Moreno-León *et al.*, 2015), una aplicación de código libre que permite analizar de forma simple proyectos de Scratch utilizando plug-ins de Hairball (Boe *et al.*, 2013). Tanto docentes como estudiantes pueden analizar automáticamente sus creaciones en Scratch y comprobar cómo programaron. Cuando se analizan los proyectos, Dr. Scratch informa un puntaje de PC que va de 0 a 21. Para otorgar dicho puntaje, la aplicación analiza el nivel de competencia alcanzado para cada aspecto del PC (Tabla 1). Finalmente se realiza una sumatoria del puntaje obtenido para cada aspecto del PC y se informa el nivel de alcanzado. A medida que va aumentando la puntuación, Dr. Scratch brinda más información de los proyectos produciendo con los usuarios más avanzados una retroalimentación para la mejora de sus proyectos. Si el puntaje asignado es bajo, se asume que se trata de programadores novatos.

Además del análisis de los proyectos con Dr. Scratch, se realizaron durante el desarrollo de la secuencia cuestionarios a los estudiantes donde se les consultó sobre cuestiones referentes al proceso de construcción de los proyectos en Scratch donde se incluyen aspectos cognitivos y metacognitivos respecto al PC, a la programación y a la IG.

Dichos cuestionarios de tipo cerrados fueron analizados según conceptos de PC y programación propuestos por Brennan y Resnick (2012) a los cuales se les realizó una modificación de denominación para realizar un paralelo con los conceptos analizados en Dr. Scratch. Respecto a los interrogantes referidos a la IG se realizó un análisis de tipo descriptivo para identificar cuestiones referentes a los modelos construidos por los estudiantes.

TABLA 1: NIVEL DE COMPETENCIA PARA CADA HABILIDAD DEL PC (MODIFICADO DE MORENO-LEÓN ET AL., 2015).

Habilidad del PC	Nivel de competencia			
	Ninguno (0)	Básico (1)	En desarrollo (2)	Competente (3)
Paralelismo	-	Dos programas en "bandera verde"	Dos programas en "tecla presionada", dos programas en "al presionar" el mismo objeto	Dos programas en "cuando reciba mensaje", crear clon, dos programas en "cuando % es >%", dos programas en "cuando el escenario cambie a"
Pensamiento lógico	-	Si	Si - Sino	Operaciones lógicas
Control de flujo	-	Secuencia de bloques	Repetir, por siempre	Repetir hasta
Interactividad con el usuario	-	Bandera verde	Tecla presionada, objeto presionado, preguntar y esperar, bloques de operaciones con ratón	Cuando % es >%, video, audio
Representación de la información	-	Modificadores de propiedades de objetos	Operaciones con variables	Operaciones con listas
Abstracción	-	Más de un programa y más de un objeto	Definición de bloques propios	Uso de clones
Sincronización	-	Esperar	Enviar, cuando reciba mensaje, parar todos, parar programas, parar programas del objeto	Esperar hasta, cuando el escenario cambie a, enviar y esperar

RESULTADOS

Las actividades fueron planteadas para que los estudiantes, a través de la construcción de proyectos con Scratch y poniendo en juego habilidades del PC, construyan modelos referentes a la IG. A continuación se presentan los resultados más relevantes.

Caracterizando las percepciones de los estudiantes en lo que refiere a sus proyectos finales de Scratch, el 56% de ellos considera que debe mejorar "mucho" el trabajo colaborativo con su compañero de grupo mientras que el 38% considera que debe mejorar dicho aspecto "poco o nada", el restante 6% no respondió a este ítem. Respecto a completar el proyecto para hacerlo funcionar correctamente, el 61% señaló que deben trabajar "mucho" sobre tal aspecto, mientras que el 33% considera que debe hacerlo "poco o nada". Como en el ítem anterior, el 6% no respondió a esta

consigna al igual que sucedió en todas las consignas. Cuando se les consultó sobre el grado de mejora en relación a organizar los proyectos para que sea claro y fácil interactuar con ellos, un 39% se inclinó por la opción "mucho", un 55% optó por las opciones "poco o nada". En referencia a la representación de los conceptos biológicos en los proyectos (ADN, ADN_r, gen de interés, vectores, plásmidos, entre otros), el 44% considera que debe mejorar "mucho" dicha representación mientras que un 50% considera que debe hacerlo "poco o nada". Siguiendo con dichos conceptos biológicos, un 39% considera que se debe mejorar "mucho" las abstracciones y las simulaciones para elaborar los proyectos mientras que el 55% considera que debe mejorar "poco o nada" dichas abstracciones y simulaciones. En relación a la comprensión del uso de los bloques, el 28% de los estudiantes manifestó que debe seguir trabajando "mucho" sobre este aspecto mientras que el 66% optó por "poco o nada". El seguir trabajando respecto a pensar sistemáticamente los pasos a seguir para crear los códigos, el 33% señaló la opción "mucho" y el 61% se inclinó por "poco o nada". Finalmente, respecto a la ejecución paralela, el 22% señaló que debe seguir trabajando "mucho" mientras que el 72% se inclinó por las opciones "poco y nada".

Cuando los mismos interrogantes fueron planteados pero respecto al análisis de los proyectos de sus compañeros, en la mayoría de los ítems la opción "mucho" fue señalada con mayores porcentajes lo que demostraría una revisión en mayor profundidad hacia los proyectos de sus compañeros que hacia los propios.

En relación a los proyectos creados por los estudiantes en Scratch, los mismos se encuentran en el nivel "básico" (56%) y en el nivel "medio" (44%) según el análisis de Dr. Scratch (Fig. 1). Si bien no se alcanzaron los niveles superiores, se destaca que fue el primer acercamiento que tuvieron dichos estudiantes con la programación y que de haber tenido más tiempo para desarrollar la secuencia didáctica los resultados hubiesen mejorado sustancialmente.



Fig. 1: Análisis de la aplicación Dr. Scratch (Moreno-León et al., 2015) realizado a dos proyectos construidos por estudiantes.

Respecto a las habilidades del PC se encontró un gradiente de desarrollo por parte de los estudiantes de las mismas (Fig. 2). La puntuación que se podía obtener por habilidad según el análisis de Dr. Scratch se encuentra en el

intervalo de 0 a 3 constituyéndose la sumatoria total de 21 en caso de obtener el puntaje máximo en cada habilidad. Para dicho análisis se contó con 16 proyectos de Scratch. La “interactividad con el usuario”, que se relaciona con la forma en qué interactúan los potenciales usuarios del proyecto, fue la habilidad más desarrollada donde los niveles “básico” y “en desarrollo” se encontraron en una frecuencia de 6 y 9 respectivamente donde se concentran el 94% de los estudiantes. Aquí utilizaron los bloques “al presionar bandera verde”, “preguntar y esperar” y “respuesta” como otras maneras de interacción. La habilidad de PC que le sigue a la anterior respecto a su desarrollo es el “paralelismo”, entendido este como la capacidad de que ocurran sucesos simultáneamente, para tal fin se utilizaron los bloques “al presionar bandera verde” y “al presionar tecla” en el mismo objeto. En este caso los niveles “básico” y “en desarrollo” se encontraron en una frecuencia de 6 y 5 respectivamente lo que representa al 69% de los estudiantes. La “sincronización” se relaciona con la forma en que los objetos o sprites actúan en relación con los demás objetivos. Aquí los estudiantes utilizaron los bloques “esperar...segundos”, “enviar a todos”, “al recibir...” y “esperar hasta que...”. Los niveles de competencia se encontraron con las frecuencias de 7 (“básico”), 4 (“nulo” y “competente”) y 1 (“en desarrollo”). En esta habilidad es donde se observó una mayor dispersión de las frecuencias.

En relación al uso de estructuras de repetición de bloques “por siempre” y “repetir hasta que” caracterizan a la habilidad “control de flujo” el nivel “básico” es el que presentó mayor frecuencia (13) representando al 81% de los estudiantes. Respecto a la “representación de la información”, el nivel “básico” se encontró con una frecuencia de 15 (94% de los estudiantes) donde los niveles “en desarrollo” y “competente” no se desarrollaron.

Luego sigue, en relación al desarrollo alcanzado por los estudiantes en los proyectos, la abstracción que evalúa cómo son utilizados y reutilizados los códigos teniendo aquí los usuarios que definir bloques propios y utilizar clones. Respecto a esta habilidad, en las categorías “nulo” (3) y “básico” (13) se encuentran el 100% de los participantes.

Por último, respecto al “pensamiento lógico” el nivel “nulo” se encontró con una frecuencia del 15 (94%)”, es decir se utilizaron en baja frecuencia los bloques del tipo “si”, “si...sino” complementados con operadores booleanos.

Los resultados anteriormente descriptos coinciden con los mostrados por Meerbaum-Salant y colaboradores (2013) que señalan que en los estudiantes participantes en cursos de Scratch se produjo un aprendizaje significativo de muchas, pero no de todas, las habilidades que caracterizan al PC. En esta línea, revisiones muestran que la utilización de Scratch favorece el aprendizaje de la programación (Moreno-León, Robles y Román-González, 2015) y el desarrollo del PC, que, como indica Wing (2011) permite el diseño y análisis de problemas y sus soluciones. En este sentido, el equipo de Scratch (2015) señala que los estudiantes al utilizar Scratch para programar aprenden a, como se indicó anteriormente, a resolver problemas de manera colaborativa al tiempo que comienzan a desarrollarse como pensadores computacionales.

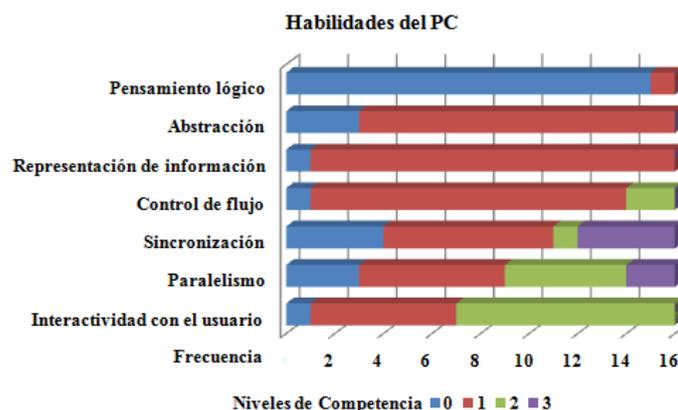


Fig. 2: Frecuencias del Nivel de Competencia alcanzado, para cada habilidad del PC, por los estudiantes en la secuencia didáctica.

En cuanto a las habilidades del PC que fueron escasamente desarrolladas, como la abstracción y el pensamiento lógico, puede deberse a que se requiere de más información para poder realizar la abstracción de ciertos conceptos, biológicos en este caso (Meerbaum-Salant *et al.*, 2013)

Como señala Wing (2008), la abstracción ocupa el nivel más alto dentro del PC lo que implica que su desarrollo requiera de más tiempo y del involucramiento de actividades cognitivas superiores.

Con el propósito de consultar a los estudiantes respecto a conceptos de PC y de programación, se plantearon situaciones en donde debían elegir el código de Scratch que mejor representara a cada afirmación, es decir a cada concepto de PC y de programación (Brennan y Resnick, 2012). El concepto de “control de flujo” (iteración o looping) fue reconocido correctamente por el 65% de los estudiantes. Por su parte, la “sincronización” (coordinación de eventos) fue identificada correctamente por el 62%. En lo relativo al “paralelismo” (ejecución paralela), el 59% distinguió la opción correcta. Los siguientes dos conceptos no se encuentran incluidos en el análisis que realiza la aplicación Dr. Scratch pero se consideran necesarios de incluir dado que se corresponden con los conceptos que más dificultades les presentan a los estudiantes. Al respecto, las “secuencias” fueron reconocidas por el 44% de los estudiantes y finalmente el 38% escogió correctamente el código que definía a los “condicionales”. Estos resultados están en línea con los mostrados en el análisis con Dr. Scratch donde en ambos casos las habilidades más sencillas son más rápidamente construidas por los estudiantes mientras que las más complejas requerirían de más tiempo y del aporte de otros procesos para su elaboración.

Por último, cuando se analizó el reconocimiento de procesos biológicos representados mediante códigos en Scratch, el 73% de los estudiantes reconoció acertadamente la “replicación del ADN” mientras que la “síntesis de proteínas” lo fue en un 47%. En lo que respecta específicamente a la IG, es decir al proceso de “construcción de una molécula de ADN recombinante”, los estudiantes lo identificaron en el 71% de los casos. Estos resultados indicarían que el uso de la programación y el planteo de secuencias didácticas que promuevan el desarrollo del PC podrían ser considerados a la hora de diseñar estrategias de aprendizaje en Biología. El PC ya fue incluido en las

investigaciones de la mayoría de las disciplinas científicas (Wing, 2011) siendo la investigación en la enseñanza de las ciencias un nicho donde comenzar a investigar dicho pensamiento.

CONCLUSIONES

A partir de los resultados obtenidos se puede concluir que los estudiantes que participaron de la secuencia pudieron desarrollar habilidades del PC al tiempo que modelizaban conceptos de IG. En este sentido Scratch demostró ser una plataforma viable para el desarrollo del PC pero se requiere de un acompañamiento cercano y efectivo (Meerbaum-Salant *et al.*, 2013). Además, suponiendo que el PC se incorpore en la enseñanza compone un reto educativo planteando interrogantes del tipo ¿Cómo se debería aprender dicho pensamiento?, ¿Cómo y cuándo debe enseñarse? (Wing, 2008). De hecho, el PC ya fue incluido en las investigaciones de la mayoría de las disciplinas científicas (Wing, 2011) siendo la investigación en la enseñanza de las ciencias un nicho donde comenzar a investigar dicho pensamiento.

En lo que respecta a las actitudes y conocimientos que poseen los estudiantes respecto a los procesos biotecnológicos, los mismos están directamente influenciados por sus propias concepciones, la sociedad y las que brindan los medios de comunicación (de la Vega Naranjo *et al.*, 2018) siendo su aprendizaje en contextos educativos una manera de formar ciudadanos con capacidades para tomar decisiones de naturaleza local reflexivas.

AGRADECIMIENTOS

Secretaría de Ciencia y Tecnología (SECyT). Universidad Nacional de Córdoba.

Miembros del grupo EDUCEVA-CienciaTIC. Departamento de Enseñanza de la Ciencia y la Tecnología. Facultad de Ciencias Exactas, Físicas y Naturales. Universidad Nacional de Córdoba.

REFERENCIAS

- [1] Adúriz-Bravo, A. e Izquierdo-Aymerich, M. (2009). "Un modelo de modelo científico para la enseñanza de las ciencias naturales". *Revista Electrónica de Investigación en Educación en Ciencias*, 4, 40-49.
- [2] Adúriz-Bravo, A. (2012). "Algunas características clave de los modelos científicos relevantes para la educación química". *Educación Química*, 23, 1-9.
- [3] Aragón, L., Jiménez-Tenorio, N., Oliva-Martínez, J. M., y Aragón-Méndez, M. M (2018). "La modelización en la enseñanza de las ciencias: criterios de demarcación y estudio de caso". *Revista Científica*, 2(32), 193-206.
- [4] Astudillo, G. J., Bast, S. G. y Willging, P. A. (2016). "Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación". *Virtualidad, Educación y Ciencia*, 7(12), 125-142.
- [5] Brennan, K. y Resnick, M. (2012). "New frameworks for studying and assessing the development of computational thinking". *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*. Vancouver (Canadá).
- [6] -Bocconi, S., Chiocciariello, A., Dettori, G., Ferrai, A. y Engelhardt, K. (2016). *Developing Computational Thinking in Compulsory Education. Implications for policy and practice*. Publications Office of the European Union, Luxembourg.
- [7] Boe, B., Hill, C., Len, M., Dreschler, G., Conrad, P., & Franklin, D. (2013). "Hairball: Lint-inspired static analysis of scratch projects". *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, ACM, Denver (EE.UU), 215-220.
- [8] Caballero, M. (2008). "Algunas ideas del alumnado de secundaria sobre conceptos básicos de genética". *Enseñanza de las Ciencias*, 26 (2), 227-243.
- [9] Confrey, J. (2006). "The evolution of design studies as methodology". En: *The Cambridge Handbook of the Learning Sciences*, 135-152, Cambridge University Press, Nueva York.
- [10] de la Vega Naranjo, M., Lorca Marín, A. A. y de las Heras Pérez, M. A. (2018). "Conocimientos y actitudes hacia la biotecnología en alumnos de último curso de Educación Secundaria Obligatoria". *Revista Eureka sobre Enseñanza y Divulgación de las Ciencias*, 15(5), 3301-3314.
- [11] Diez de Tancredi, D. y Caballero, C. (2004). "Representaciones externas de los conceptos biológicos de gen y cromosoma. Su aprendizaje significativo". *Revista de Investigación*, 56, 91-121.
- [12] Gilbert, J. K. (2002). "Moving between the modes of representation of a model in science education: Some theoretical and pedagogic implications". *Conference Philosophical, Psychological, Linguistic Foundations for Language and Science Literacy Research*. Columbia Británica (Canadá).
- [13] Gilbert, J. K. y Justí, R. (2016). *Modelling-based Teaching in Science Education*. Springer International Publishing, Suiza.
- [14] Gómez Galindo, A. A. (2014). "El uso de representaciones multimodales y la evolución de los modelos escolares". En: *Avances en Didáctica de la Química: Modelos y Lenguajes*, 51-61, Ediciones Universitarias de Valparaíso, Valparaíso (Chile).
- [15] Gretter, S. y Yadav, A. (2016). "Computational Thinking and Media and Information Literacy: An Integrated Approach to Teaching Twenty-First Century Skills". *TechTrends*, 60, 510-516.
- [16] Gouws, L. A., Bradshaw, K., y Wentworth, P. (2013). "Computational thinking in educational activities: An evaluation of the educational game light-bot". *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, ACM, Canterbury (UK), 10-15.
- [17] ISTE y CSTA. (2011). Operational Definition of Computational Thinking for K-12 Education. National Science Foundation under Grant. CNS- 1030054.
- [18] Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J. y Werner, L. (2011). "Computational thinking for youth in practice". *ACM Inroads*, 2(1), 32-37.
- [19] Lehrer, R. y Schauble, L. (2006). "Scientific thinking and science literacy: Supporting development in learning in contexts", En: *Handbook of Child Psychology, Child psychology in practice*, 153-196, John Wiley and Song, Nueva York.
- [20] Lewis, J. y Wood-Robinson, C. (2000). "Genes, chromosomes, cell division and inheritance: do students see any relationship?" *International Journal of Science Education*, 22(2), 177-195.
- [21] Lee, L. S. y Koh, J. H. (2014). "Review on Teaching and Learning of Computational Thinking through Programming: What is Next for K-12?" *Computers in Human Behavior*, 41, 51-61.
- [22] Malan, D. J., y Leitner, H. H. (2007). "Scratch for budding computer scientists". *ACM SIGCSE Bulletin*, 39(1), 223-227.
- [23] Manovich, L. (2013). *Software takes command*. Bloomsbury, Nueva York.
- [24] Meerbaum-Salant, O., Armoni, M. y Ben-Ari, M. (2013). "Learning computer science concepts with Scratch". *Computer Science Education*, 23(3), 239-264.
- [25] Moreno-León, J., Robles, G. y Román-González, M. (2015). "Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking". *Revista de Educación a Distancia*, 46.
- [26] National Research Council. (2010). Committee for the Workshops on Computational Thinking: Report of a workshop on the scope and nature of computational thinking. National Academies Press, Washington DC.

- [27] Occelli, M., Malin Vilar, T. y Valeiras, N. (2011). "Conocimientos y actitudes de los estudiantes de la ciudad de Córdoba (Argentina) en relación a la Biotecnología". *Revista Electrónica de Enseñanza de las Ciencias*, 10(2), 227-242.
- [28] Oliva, J. M. y Aragón, M. M. (2009). "Contribución del aprendizaje con analogías al pensamiento modelizador de los alumnos en ciencias: marco teórico". *Enseñanza de las Ciencias*, 27(2), 195-208.
- [29] Pedrancini, V. D., Corazza-Nunes, M. J., Bellanza Galuch, M. T., Olivo Rosas Moreira, A. L. y Ribeiro, A. C. (2007). "Ensino e aprendizagem de Biologia no ensino médio e a apropriação do saber científico e biotecnológico". *Revista Eletrônica de Enseñanza de las Ciencias*, 6(2), 299-309.
- [30] Pedrancini, V. D., Corazza-Nunez, M. J. Bellanza Galuch, M. T. Olivo Rosas Moreira, A. L. y Carvalho Nunes, W. M. (2008). "Saber científico y conocimiento espontáneo: opiniões de alunos do ensino médio sobre transgênicos". *Ciência y Educação*, 14(1), 135-146.
- [31] Resnick, M. (2013). "Learn to code, code to learn". *EdSurge, Education Technology News and Resources*.
- [32] Rinaudo, M. y Donolo, D. (2010). "Estudios de diseño. Una alternativa prometedora en la investigación educativa". *Revista de Educación a Distancia*, 22.
- [33] Roa Acosta, R. y Valvueda Ussa, E. O. (2013). "Incurción de la biotecnología en la educación: tendencias e implicaciones". *Revista Colombiana de Biotecnología*, 15(2), 156-166.
- [34] Román-González, M. (2015). "Test de Pensamiento Computacional: principios de diseño, validación de contenido y análisis de ítems". En: *Perspectivas y avances de la investigación*, 279-302, UNED, Madrid (España).
- [35] Rushkoff, D. (2010). *Program or be programmed*. OR Books, New York.
- [36] Sunkel, G. y Trucco, D. (2012). *Las tecnologías digitales frente a los desafíos de una educación inclusiva en América Latina. Algunos casos de buenas prácticas*. Comisión Económica para América Latina y el Caribe (CEPAL), Publicación de las Naciones Unidas, Santiago de Chile (Chile).
- [37] Tegegne, F., Aziz, A., Bhavsar, H., y Wiemers, R. (2013). "Awareness of and attitudes towards biotechnology by Tennessee State university students with different backgrounds and majors". *Journal of Biotechnology Research*, 5, 16-23.
- [38] Torres, J. y Vasconcelos, C. (2017). "Desarrollo y validación de un instrumento para analizar las visiones de los profesores sobre modelos". *Revista Eureka sobre Enseñanza y Divulgación de las Ciencias*, 14(1), 181-198.
- [39] Valverde Berrocoso, J., Fernández Sánchez, M. R. y Garrido Arroyo, M. C. (2015). "El pensamiento computacional y las nuevas ecologías del aprendizaje". *Revista de Educación a Distancia*, 46(3).
- [40] Vázquez-Cano, E. y Ferrer Delgado, D. (2015). "La creación de videojuegos con Scratch en educación secundaria". *Communication papers. Media literacy & Gender Studies*, 4(6), 63-73.
- [41] Venville, G. y Donovan, J. (2008). "How pupils use a model for abstract concepts in genetics". *Journal of Biological Education*, 43(1), 6-14.
- [42] Wing, J. M. (2006). "Computational thinking". *Communications of the ACM*, 49(3), 33-35.
- [43] Wing, J. M. (2008). "Computational thinking and thinking about computing". *Philosophical Transactions of the Royal Society a Mathematical Physical and Engineering Sciences*, 366(1881), 3717-3725.
- [44] Wing, J. M. (2011). "Research notebook: Computational Thinking. What and Why?" The Link Magazine, Spring, Carnegie Mellon University School of Computer Science, Pittsburgh.