



Revista Iberoamericana de Tecnología en Educación y
Educación en Tecnología

ISSN: 1851-0086

ISSN: 1850-9959

editor-teyet@lidi.info.unlp.edu.ar

Universidad Nacional de La Plata

Argentina

Fussero, Gimena B.; Ocelli, Maricel; Chiarani, Marcela
Pensamiento Computacional y aprendizaje de la Ingeniería
Genética: Una aproximación a través de una investigación de diseño
Revista Iberoamericana de Tecnología en Educación y Educación
en Tecnología, núm. 30, 2021, Julio-Diciembre, pp. 40-50
Universidad Nacional de La Plata
Argentina

DOI: <https://doi.org/10.24215/18509959.30.e4>

- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org



Pensamiento Computacional y aprendizaje de la Ingeniería Genética: Una aproximación a través de una investigación de diseño

Computational Thinking and Genetic Engineering learning: An Approach from Design Research

Gimena B. Fussero¹, Maricel Occelli², Marcela Chiarani³

¹ 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

² Grupo 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

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

gimenafussero@unc.edu.ar, maricel.occelli@unc.edu.ar, mcchiarani@gmail.com

Recibido: 25/03/2020 | **Corregido:** 10/12/2020 | **Aceptado:** 29/05/2021

Cita sugerida: G. B. Fussero, M. Occelli, M. Chiarani, "Pensamiento Computacional y aprendizaje de la Ingeniería Genética: Una aproximación a través de una investigación de diseño," *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, no. 30, pp. 40-50, 2021. doi: 10.24215/18509959.30.e4

Esta obra se distribuye bajo **Licencia Creative Commons CC-BY-NC 4.0**

Resumen

La programación se está incorporando en los espacios educativos a través de diferentes resoluciones. Una de las perspectivas de su incorporación es hacerlo de manera transversal conjuntamente con otros espacios curriculares y en este contexto el Pensamiento Computacional adquiere relevancia. En este trabajo presentamos una investigación que adoptó la metodología conocida como estudios de diseño donde se planificó una secuencia didáctica para el aprendizaje de la Ingeniería Genética, en la escuela secundaria, la cual se encuentra prescripta en los diseños curriculares de 5° año con orientación en Ciencias Naturales. Para ello se le propuso a las y los estudiantes situaciones problemáticas en donde debían construir modelos de Ingeniería Genética utilizando Scratch. Los principales resultados muestran que la utilización de Scratch como primer lenguaje de programación permite el acercamiento del estudiantado al desarrollo del Pensamiento Computacional al tiempo que desarrollan prácticas científicas como la modelización.

Palabras clave: Programación; Scratch; Modelización; Didáctica; Biología; Escuela secundaria.

Abstract

Programming is being incorporated into educational spaces through different resolutions. One of the perspectives of this incorporation is by means of a transversal relationship with other curricular spaces and in this context Computational Thinking acquires relevance. In this work we present an investigation that adopted the methodology known as design studies where a didactic sequence was planned for the learning of Genetic Engineering in high schools, which is included in the 5th year programme with an orientation in Natural Sciences. This end, different problematic situations were proposed to students where they had to build Genetic Engineering models using Scratch. The main results show that the use of Scratch as the first programming language allows students to approach Computational Thinking while developing scientific practices such as modeling.

Keywords: Programming; Scratch; Modeling; Didactic; Biology; High school.

1. Introducción

Los recursos tecnológicos pueden ampliar las capacidades de razonar, crear y expresarse cuando su integración en procesos educativos no tiene como propósito principal la operación de herramientas digitales sino la apropiación de diversos modos de pensar y crear asociados a la tecnología [1]. En este escenario, la programación y el Pensamiento Computacional (PC) cobran relevancia en las discusiones tecnoeducativas [2]. Respecto a la programación hace varios años se ha incorporado en los ámbitos escolares y en el último tiempo su adhesión ha aumentado por diferentes razones. Por un lado, porque posiciona a las y los estudiantes frente a situaciones problemáticas cuya resolución requiere de procesos cognitivos complejos y de la elaboración de representaciones permitiendo el desarrollo de competencias como lo son el pensamiento crítico, la comunicación efectiva, el trabajo en equipo y el manejo de información [3]. Por el otro, si se diseñan actividades apropiadas, la programación promueve la generación de nuevas ecologías de aprendizaje [4]. En otras palabras, la programación se está incluyendo en los diferentes niveles educativos ya que no se trata solo de una competencia cognitiva que se utiliza para diseñar códigos sino también de una competencia social requerida para participar en grupos [5], [4] y uno de los mayores grupos de programación lo representa Scratch. Este lenguaje se basa en un conjunto de instrucciones icónico/textuales que se combinan para crear programas permitiendo avanzar sobre la resolución de problemas, el análisis, la síntesis, la conceptualización, el manejo de información, el pensamiento sistémico, el pensamiento crítico y la metacognición [2]. Otra ventaja que ofrece Scratch es que ayuda al aprendizaje mediante el desarrollo del PC [6].

A pesar de que el concepto de PC fue propuesto en el año 2006 por Wing, a la fecha existen diferentes perspectivas, interpretaciones y posicionamientos. Aquí se considerará al PC según la definición propuesta por Wing [7] quien afirma que "El PC es el proceso de pensamiento involucrado en la formulación de un problema y la expresión de su(s) solución(es) de tal manera que pueda ser abordada de forma efectiva por un agente-procesador de la información, como una computadora, un ser humano o una combinación".

Considerando las potencialidades que brindan los lenguajes de programación, y más específicamente Scratch, surgió el interés de incorporar esta herramienta en el diseño de secuencias didácticas de Biología que impliquen la construcción de modelos sobre conceptos de Ingeniería Genética (IG) la cual se está convirtiendo en una de las revoluciones científicas más importantes del siglo XXI [8].

Entonces, dado que los lenguajes de programación proporcionan entornos de modelización y que la IG requiere de la construcción de modelos para su comprensión se propuso a Scratch como mediador de la modelización. En este trabajo se estudió qué habilidades

del PC fueron puestas en juego por el estudiantado al momento de construir modelos de IG en Scratch.

2. Marco teórico

2.1. La programación en los escenarios educativos

La interactividad se encuentra presente en todas partes, desde videojuegos hasta cerraduras electrónicas, incluyendo aplicaciones de teléfonos móviles, computadoras personales, entre otras. Así como se valoran las habilidades para consumir (leer) y para crear (escribir) un texto, la capacidad de consumir medios interactivos debe ir acompañada de habilidades complementarias para crearlos [9]. 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 [10].

Sumado a lo anterior y dado el creciente rol que ocupa la informática en la educación, la comprensión de cómo las personas interactúan con las computadoras y entrenan el abordaje algorítmico a los problemas a través de lenguajes informáticos se convirtió en un área de interés para las investigaciones en educación [11]. Sin embargo, no existe un consenso de cómo deben incluirse estos contenidos en el sistema educativo [12]. En los últimos años surgieron iniciativas que invitan al estudiantado, desde edades tempranas, a acercarse a la programación y al PC [13]. Actividades de este tipo se encuentran en todos los continentes con dos enfoques muy diferentes en relación a los objetivos y a la metodología con que dichos contenidos son introducidos en los currícula. Inglaterra y Eslovaquia, por ejemplo, propusieron una asignatura específica centrada en el desarrollo de habilidades informáticas mientras que Estonia o Finlandia optaron por utilizar estas habilidades como un instrumento para el aprendizaje de otras asignaturas [12]. Estas propuestas consideran a la programación como una actividad que permitiría el desarrollo de competencias relacionadas con la realidad de las y los estudiantes [3].

En Argentina, a través de la Resolución Ministerial N° 1410/2018 se incorpora la programación y la robótica a la educación obligatoria. No se instalan como fin último sino que se consideran saberes para desarrollar la alfabetización digital. La JISC (*Joint Information Systems Committee*) define a la alfabetización digital como aquellas capacidades que les permiten a las personas vivir, aprender y desempeñarse en una sociedad digital. En el caso de la educación secundaria uno de los objetivos de la inclusión de la programación hace referencia a la utilización del PC para desarrollar proyectos de robótica o programación física, de modo autónomo, crítico y responsable, construyendo soluciones originales a problemas del entorno social, económico y ambiental del estudiantado.

En este contexto el construccionismo cobra relevancia [6].

Seymour Papert, quien participó en el desarrollo del lenguaje de programación Logo en el MIT en la década de 1960, en su libro *Mindstorms* (1980) explicaba que "al programar una computadora, los niños adquieren un sentido de maestría frente a uno de los tipos de tecnología más potente y novedosa, y establecen un contacto íntimo con algunas de las ideas más profundas de las ciencias, las matemáticas y la construcción de modelos intelectuales". Bajo este paradigma, se considera que el aprendizaje y la creatividad se mejoran a través de la interacción con pares ya que se trata de procesos sociales [9]. En este sentido, el uso de herramientas digitales y entornos virtuales permiten que niñas y niños, cuando se encuentran utilizando una computadora, busquen asistencia en sus pares aumentando la socialización [4].

Además, aunque gran cantidad de jóvenes interactúan con medios digitales todo el tiempo, solo una escasa parte son capaces de crear sus propios juegos, animaciones, o simulaciones, es como si pudieran "leer" pero no "escribir" [14]. Otras investigaciones muestran que, cada vez más jóvenes actúan regularmente como creadores de contenido y programadores informales ya sea modificando el timbre de un smartphone o creando contenido para un sitio web [15]. Sin embargo, modificar el timbre de un teléfono celular no es lo mismo que escribir y comprender un algoritmo por lo que las y los jóvenes requieren de nuevas competencias.

2.2. El Pensamiento Computacional

El concepto de PC fue propuesto en el año 2006 por Jeanette Wing (Seymour Papert realizó una primera aproximación en el año 1980), sin embargo a la fecha existen diferentes perspectivas e interpretaciones no existiendo en la literatura especializada una definición clara [2].

Luego, Wing [7] realizó una revisión de la definición original enunciando que "El PC incluye los procesos de pensamiento implicados en el diseño de problemas y de sus soluciones de tal modo que queden representados de una forma en que puedan ser resueltos de manera eficaz por un agente procesador de la información".

En el año 2009 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 implica diferentes características. Destacan que el PC es un enfoque para la resolución de problemas que posibilita la integración de las tecnologías digitales con las ideas humanas no reemplazando la creatividad y el pensamiento crítico sino reforzando dichas habilidades [16].

Posteriormente para aportar a la discusión, Wing [17] planteó modificaciones a la primera definición. Luego, en

el año 2012, la *Royal Society* propuso que "El PC es el proceso de reconocer los aspectos de la computación en el mundo que nos rodea, y aplicar herramientas y técnicas de la informática para comprender y razonar sobre sistemas y procesos naturales y artificiales" [18]. El equipo de Scratch del MIT, en el 2015, definió al PC como "Un conjunto de conceptos, prácticas y perspectivas que se basan en las ideas del mundo de la informática". Resnick y Rusk [9] expresan que el PC no es simplemente una forma de aprender habilidades de resolución de problemas, sino también una herramienta que permite expresarse con medios digitales.

A pesar de tratarse de un concepto emergente, la mayoría de los posicionamientos siguen a la línea primigenia propuesta por Wing [1]. Relacionado a lo anterior y ante la falta de una definición única respecto al PC, una serie de conceptos y habilidades centrales emergen una y otra vez de la literatura. Estos incluyen: abstracción, pensamiento algorítmico, automatización, descomposición y depuración [12]. En este trabajo se consideraron las habilidades propuestas por Moreno-León y colaboradores [19]: Paralelismo, Pensamiento lógico, Control de flujo, Interactividad con el usuario, Representación de la información, Abstracción y Sincronización. A continuación se realiza una breve descripción de cada una de ellas:

-Paralelismo: Consiste en programar secuencias de instrucciones para que sucedan simultáneamente. En este sentido, las computadoras realizan lo que se les ordena y son capaces de realizar muchas acciones en poco tiempo, incluso pueden realizar varias al mismo tiempo [6].

-Control de flujo: Radica en tomar decisiones con base a ciertas condiciones, que apoyan la expresión de múltiples resultados [6]. Permite determinar el orden en el que los objetos realizan diferentes acciones en determinados momentos.

-Interactividad con el usuario: Reside en la creación de programas que requieran el ingreso de datos o la toma de decisiones por parte del operador. Insta a pensar previamente qué aspectos se quieren dejar a elección de las usuarias y los usuarios para que interactúen con el programa.

-Representación de la información: Se trata de establecer, en cada proyecto, un conjunto de información sobre los personajes o componentes para que se pueda ejecutar correctamente. Por ejemplo, se necesitan establecer, para luego conocer, la posición y el tamaño de los personajes como así también otra serie de atributos que en cada momento de la ejecución del proyecto tienen un valor que puede ser modificado por la programación.

-Abstracción: Abstraer consiste en dividir un problema grande en problemas más pequeños y de fácil solución. Permite lograr que un problema, artefacto o proceso sea más comprensible mediante la reducción de detalles innecesarios. Reside en elegir el detalle correcto de manera que el problema se vuelva más fácil de

comprender, sin perder información importante. La abstracción se encuentra relacionada con la descomposición del problema ya que es una forma de pensar sobre los sistemas en términos a sus componentes [20].

-Sincronización: Se basa en establecer un orden determinado, por ejemplo, que un personaje realice una acción cuando otro finaliza conformándose de esta manera una cadena ordenada de acciones. Esta habilidad se encuentra relacionada con el pensamiento algorítmico y la secuenciación ya que estos implican llegar a una solución a través de una definición clara de pasos ordenados a seguir para la resolución de un problema [20].

-Pensamiento lógico: Permite reconocer el problema que se intenta solucionar y buscar una solución que se pueda programar analizando los resultados obtenidos a través de la modificación de variables (entradas). Es importante poder anticipar y programar teniendo en cuenta todas estas situaciones [20].

En otros ámbitos, el PC se utiliza directa o indirectamente y plantea retos educativos. Al respecto, existe una tendencia para incorporar al PC a la educación obligatoria para contribuir a la formación de estudiantes con habilidades para ser participantes con una actitud activa y reflexiva en la cultura digital [21]. Por ello ha resurgido en los últimos años un movimiento educativo a nivel internacional relacionado con la introducción del PC, la programación y la robótica en las escuelas [4].

2.3. El lenguaje de programación Scratch

Scratch se desarrolló en el *Lifelong Kindergarten del Media Laboratory* del MIT. Se trata de un entorno de programación que permite la creación de proyectos en forma de historias interactivas, juegos y secuencias animadas con el agregado que dichas creaciones son compartidas en la web [3]. Desde su lanzamiento en mayo de 2007, el sitio web de Scratch (<http://scratch.mit.edu>) se ha convertido en una comunidad en línea con personas compartiendo, discutiendo y remezclando proyectos [14]. Scratch se constituyó en una de las más grandes comunidades de aprendizaje siendo utilizado masivamente en todo el mundo con más de siete millones de usuarias y usuarios y más de 10 millones de proyectos compartidos [19]. Uno de los principales objetivos de Scratch es que la programación se incluya en los escenarios educativos para el desarrollo de habilidades y para mejorar el aprendizaje en otras disciplinas [14]. Al respecto, fue incluido en escuelas primarias [19], en escuelas secundarias [22], en universidades [9] y en actividades extracurriculares [15].

Scratch tiene sus orígenes en el lenguaje de programación Logo creado en la década de 1960 con el propósito de introducir a las niñas y los niños en el uso de las computadoras como instrumentos para el aprendizaje. Años más tarde, con el surgimiento de las computadoras personales, el interés por la programación creció pero luego disminuyó. Resnick y colaboradores [14] indican

que los siguientes factores pudieron incidir en dicho desinterés inicial: los lenguajes de programación iniciales eran demasiado difíciles de usar y gran cantidad de niñas y niños simplemente no podían dominar la sintaxis de la programación; la programación se introdujo con actividades que no estaban conectadas con los intereses de las y los jóvenes y finalmente la programación fue introducida en contextos donde nadie podía proporcionar orientación. En este escenario surge Scratch el cual fue diseñado por Resnick y su equipo. Para cumplir con los objetivos de Scratch, sus creadores establecieron tres principios básicos: que estimule la creatividad, que sea más significativo y más social que otros entornos de programación. Según Resnick y Rusk [9] Scratch está siendo introducido en el currículum de diversos países porque se fomenta el espiral del pensamiento creativo promoviendo diferentes aprendizajes en el estudiantado. En este sentido, varias investigaciones muestran resultados favorables en relación a la inclusión de Scratch en ámbitos educativos.

Erol y Kurt [23] indicaron que Scratch no sólo motiva a las y los estudiantes en su primera incursión en la programación sino que también familiariza a personas inexpertas con los fundamentos de la programación sin la distracción de la sintaxis.

Por su parte Calder [24] señaló que Scratch demostró ser un espacio atractivo y relativamente fácil de usar para la resolución de problemas al tiempo que proporcionó un ambiente valioso para explorar conceptos matemáticos. Resultó ser un medio por el cual los programas eran fáciles de componer y modificar, fomentando el uso de habilidades críticas, metacognitivas y reflexivas.

En el caso de Kafai y Field [15] se centraron en caracterizar a las usuarias y los usuarios de Scratch. Encontraron que permite que se involucren con el programa con sus propios sistemas de reglas. La informalidad de Scratch en la práctica social replica los entornos de programación informales que las y los jóvenes encuentran diariamente y conocen bien como por ejemplo sus smartphones. Indicaron que Scratch es una práctica prometedora para el ingreso del estudiantado a la programación.

Por su parte Meerbaum-Salant y colaboradores [22] investigaron la utilización de Scratch y el desarrollo del PC. Encontraron que pocos proyectos incluyen bucles de repetición y variables ya que estos requieren de diferentes niveles estructurales y relacionales, es decir de mayores habilidades cognitivas. En el caso de la abstracción, considerada central en el desarrollo del PC, es uno de los conceptos más difíciles de percibir y es por este motivo que el estudiantado tiende a reducir a dicho concepto en sus proyectos. Sostienen que estas dificultades pueden ser superadas.

Otras investigaciones señalan la tendencia creciente de incluir a Scratch en las universidades en cursos de informática para principiantes para la enseñanza de habilidades y conceptos de programación debido a los

resultados positivos obtenidos en estudiantes de entre 8 y 18 años [5]. Si alguna vez migran a lenguajes basados en texto, el conocimiento de los elementos de programación, como el control de flujo y las variables, se transfiere de bloques a texto permitiendo una transición entre ambos lenguajes. En este sentido no toda la comunidad informática valora a Scratch pero reconocen que se pueden construir proyectos interesantes solo con algunos conceptos de la programación [26]. Pero enfatizar lo que le falta a Scratch es subestimar las posibilidades que ofrece [23].

En resumen, Scratch ayuda al aprendizaje mediante el PC ya que se encuentra enfocado en actividades basadas en el diseño ofreciendo un gran abanico de herramientas para ayudar a resolver dichas actividades desde una perspectiva creativa, investigativa y exploratoria [6].

2.4. Herramientas digitales en el aprendizaje de la Genética y de la IG

Los entornos multimedia de instrucción en general y las animaciones en particular tienen un gran potencial para mejorar la forma en que las personas aprenden [27]. Cuando una animación simula procesos reales que incluyen, por ejemplo, movimiento, permite que las y los estudiantes ejecuten experimentos virtuales que de otra manera no serían factibles en un laboratorio escolar y les permite enfocarse en conceptos abstractos particulares [28].

En este contexto resulta de interés incluir la utilización de animaciones en Genética especialmente cuando se enseñan procesos dinámicos. Al respecto, docentes e investigadores aconsejan el uso de animaciones computacionales u otras herramientas digitales cuando se enseñan procesos dinámicos destacando el potencial de este tipo de herramientas [29] coincidiendo en que la visualización y la modelización constituyen un componente importante en el desarrollo de la ciencia y en su enseñanza [30]. Además, posicionar a las y los estudiantes en creadores de modelos y de simulaciones les brinda un potencial cognitivo que promueve su actividad como productores y no solamente como usuarias y usuarios de este tipo de tecnologías.

3. Metodología

En esta investigación se adoptó la metodología conocida como *Estudios o Experimentos de Diseño* los cuales se orientan a generar conocimientos tendientes a mejorar las prácticas de enseñanza [31]. En otras palabras, el propósito es comprender la ecología de aprendizaje y desarrollar conocimientos que permitan analizar y transformar a otros contextos educativos [32]. Las investigaciones de diseño se caracterizan por responder a dos tipos de metas. Una meta pedagógica vinculada a los procesos que se buscan promover con la comunidad que participa del diseño y una meta teórica referida al

conocimiento que se espera construir a partir del análisis de la información recolectada. En este trabajo la meta pedagógica refiere a fomentar aprendizajes en IG a partir de su modelización. Por su parte, desde la meta teórica se busca aportar conocimiento acerca del desarrollo de las habilidades del PC que se ponen en juego al construir modelos en Scratch.

La investigación se llevó adelante en una escuela secundaria, de gestión privada, ubicada en el barrio Jardín Espinosa, al sur de la ciudad de Córdoba Capital. La secuencia didáctica diseñada se implementó en un 5° año en la asignatura Biología con orientación en Ciencias Naturales donde se tiene prescripto en los diseños curriculares a la IG. Las y los estudiantes (N=34) contaban con una edad de entre 16 y 17 años al momento de participar de las actividades. En relación a la organización de la investigación, la misma constó de tres fases: preparación, implementación y análisis [31].

La fase de preparación incluyó la elaboración colaborativa, con la docente a cargo del curso, del diseño instructivo a implementar. Se diseñó la secuencia didáctica teniendo como principal objetivo pedagógico que el estudiantado aprenda cuáles son los principales pasos para la construcción de una molécula de ADN tomando como eje una situación problemática en torno a la construcción de una molécula de insulina recombinante. En esta etapa también se construyeron los instrumentos que permitieron la recolección de los datos. Se elaboraron cuestionarios a modo de pre y post test que fueron realizados de manera individual por las y los estudiantes. Ambos cuestionarios contaban con preguntas de tipo abierto y otras de tipo semiestructuradas que hacían referencia a aspectos conceptuales y metacognitivos. Ambos cuestionarios fueron validados (analizados para determinar si son adecuados para el propósito para el cual han sido construidos) por dos expertas en metodología de la investigación educativa (quienes realizaron una validación de contenido, una validación de constructo y una validación de criterio) y por un grupo de estudiantes de 5° año, pertenecientes a otro curso de la misma institución (a través de una prueba piloto) que participaron de manera voluntaria.

La fase siguiente, la implementación del diseño, ocurrió durante 5 semanas consecutivas de clase. Durante dicho periodo se realizó un registro de campo de la dinámica de las clases. Se comenzó con la realización del pre test y luego, se propuso al estudiantado actividades para aproximarse a Scratch y para realizar un seguimiento del desarrollo de sus conocimientos. Aquí el grupo de estudiantes modelizó en proyectos (N=17) de Scratch la construcción de una molécula de insulina recombinante. Al finalizar las actividades las y los estudiantes desarrollaron el post test. Finalmente se realizaron entrevistas a los grupos focales (grupos de estudiantes a los cuales se les realiza una entrevista semiestructurada de manera grupal) y a la docente.

La última fase implicó el análisis. Para analizar las habilidades del PC desarrolladas por el grupo de

estudiantes durante la construcción de los proyectos en Scratch, se utilizó [®]Dr. Scratch [19]. Se trata de una aplicación de código libre que permite analizar de forma simple proyectos de Scratch utilizando plug-ins de Hairball. El análisis se realiza ingresando a la página web <http://www.drscratch.org/> y se coloca el link del proyecto o el proyecto descargado a analizar. Una vez colocado el link o el proyecto descargado se hace correr la aplicación y se efectúa el análisis. La información que aporta la evaluación en [®]Dr. Scratch consiste en un puntaje que va de 0 a 21. Al momento de otorgar dicho puntaje, el software analiza el nivel de competencia alcanzado para cada habilidad del PC considerada (paralelismo, pensamiento lógico, control de flujo, interactividad con el usuario, representación de la información, abstracción y sincronización).

Cada habilidad puede obtener un puntaje de entre 0 y 3 de acuerdo al desarrollo alcanzado en cada proyecto. El puntaje 0 se corresponde al nivel *nulo*, el puntaje 1 al nivel *básico*, el puntaje 2 al nivel *en desarrollo* y finalmente el puntaje 3 al nivel *competente*. Una vez otorgado el puntaje a cada habilidad se realiza una sumatoria obteniéndose un puntaje final e informando el nivel alcanzado para el proyecto analizado. Los proyectos se clasifican según el puntaje alcanzado entre 0 y 21. Los proyectos que obtienen un puntaje de entre 0 y 7 se ubican dentro del nivel *básico*, los que se encuentran con un puntaje de entre 8 y 14 se encuentran dentro del nivel *medio* y desde 15 a 21 en el nivel *alto*.

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

4. Resultados

4.1. Análisis de los proyectos de Scratch

Recuperando los proyectos realizados por el estudiantado para representar la construcción de una molécula de insulina recombinante mediante IG, el 47% de los proyectos según el análisis en [®]Dr. Scratch se encontraban en el nivel *básico* y el 53% restante en el nivel *medio* (Fig. 1 y Fig. 2).



Figura 1. Análisis realizado con [®]Dr. Scratch sobre un proyecto elaborado por estudiantes para representar la construcción de insulina recombinante. El proyecto alcanzó el nivel *medio* con un puntaje de 12. Arriba se muestra la retroalimentación que ofrece el software para mejorar el proyecto. Abajo se indica el nivel alcanzado para cada habilidad del PC considerada.



Figura 2. Capturas de pantalla de un proyecto que muestra un modelo de IG para la construcción de una molécula de insulina recombinante.

Estos resultados son coincidentes con los presentados por Meerbaum Salant [22] quienes encontraron que los niveles de rendimientos del estudiantado utilizando Scratch no siempre fueron altos y que los conceptos más abstractos requieren del involucramiento de actividades cognitivas superiores. Si bien la programación en Scratch encuentra ciertas similitudes con, por ejemplo la personalización de smartphone, se convierte en una práctica prometedora para que las y los jóvenes se conviertan en productores/diseñadores no siendo meros consumidores de entornos digitales [25].

4.2. Habilidades del PC desarrolladas

En relación a las habilidades del PC desarrolladas por el grupo de estudiantes a lo largo de su participación en las actividades se realizó un seguimiento de las mismas de manera paulatina incrementando las habilidades

registradas a medida que se complejizaban las actividades requeridas respecto a la programación.

En la instancia del pre test, las habilidades que fueron consideradas fueron la secuenciación (enunciar de manera ordenada una serie de comandos), el completamiento (completar un conjunto incompleto de comandos previamente dado) y la depuración (depurar un conjunto incorrecto de comando previamente dado). Para su concreción se tomaron ítems del test de PC propuesto por Román-González [10]. En cuanto a la secuenciación el 97% de las y los estudiantes pudo reconocerla correctamente mientras que cuando se agregaron estructuras de repetición a dicha habilidad el porcentaje fue del 67%. En relación al completamiento, el 97% del estudiantado lo hizo correctamente mientras que cuando se le agregaron estructuras de repetición y condicionales dicho porcentaje fue del 56%. En el caso de la depuración, el 65% logró realizarlo de manera correcta. Estos resultados coinciden con otras investigaciones [22] en donde se encontró que sólo en el 10% de los proyectos en Scratch se utilizan estructuras de repetición y condicionales.

Para proseguir con el análisis de las habilidades del PC se tomaron los proyectos finales que programaron las y los estudiantes en Scratch en los cuales debían modelizar la construcción de insulina recombinante utilizando IG. Para el análisis de las habilidades del PC en esta instancia se utilizó [®]Dr. Scratch [19] en donde se consideraron las siguientes habilidades: paralelismo, pensamiento lógico, control de flujo, interactividad con el usuario, representación de la información, abstracción y sincronización. Aquí se encontró un gradiente en el nivel de desarrollo de las habilidades del PC consideradas (Fig. 3).

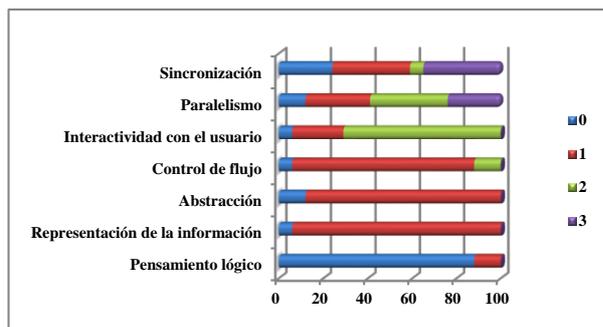


Figura 3. Nivel de desarrollo alcanzado para cada habilidad del PC consideradas en [®]Dr. Scratch de los proyectos realizados por el estudiantado.

La sincronización fue la habilidad que evidenció un mayor desarrollo siendo el nivel *competente* (3) alcanzado por el 35% del estudiantado y el nivel *en desarrollo* (2) por el 6%. Aquí se utilizaron los bloques “esperar...segundos”, “enviar a todos”, “al recibir...” y “esperar hasta que...”. La siguiente habilidad, el paralelismo, que en Scratch se recurre a la utilización de los bloques “al presionar bandera verde” y “al presionar tecla” en el mismo objeto, alcanzó en un 59% los niveles *en desarrollo* (35%) y

competente (24%). Respecto a la interactividad con el usuario, los estudiantes se desempeñaron en el nivel *en desarrollo* en el 71% de los casos. Aquí se utilizaron los bloques "al presionar bandera verde", "preguntar y esperar" y "respuesta" entre otros. El control de flujo se obtiene incorporando a los proyectos los bloques "por siempre" y "repetir hasta que" logrando estructuras de repetición. Para esta habilidad los niveles alcanzados por las y los estudiantes se concentraron con un 94% entre el grado de desarrollo *nulo* y *básico*. Respecto a la representación de la información, el 94% del estudiantado se encontró en el nivel *básico* donde no se desarrollaron modificaciones en las propiedades de los objetos ni operaciones con variables. La abstracción se evaluó respecto a cómo son utilizados y reutilizados los códigos teniendo aquí que definir bloques propios y utilizar clones. Para esta habilidad el total del estudiantado se ubicó entre los niveles *nulo* (12%) y *básico* (88%). Acerca de la representación de la información los niveles alcanzados por las y los estudiantes son similares a los conseguidos en la abstracción. En este caso en el nivel *nulo* se encontró al 6% del estudiantado mientras que en el nivel *básico* al resto. Finalmente, para el caso del pensamiento lógico, representado por la utilización de los bloques "sí", "sí...sino" complementados con operadores booleanos, no se observó un desarrollo en el 88% de los proyectos encontrándose el porcentaje restante en el nivel *básico*.

Los resultados anteriormente descriptos coinciden con los mostrados por Meerbaum Salant y colaboradores [22] quienes señalan que en estudiantes que participaron en cursos de Scratch se produjo un aprendizaje significativo de muchas, pero no de todas, las habilidades que caracterizan al PC. La diferencia en el desarrollo de las diferentes habilidades del PC puede deberse a que algunas habilidades requieren para su desarrollo de un trabajo continuo en el tiempo y a su vez implican mayores desafíos cognitivos para las y los estudiantes. En lo que respecta a las habilidades del PC que fueron desarrolladas en menor medida por el estudiantado, como la abstracción y el pensamiento lógico estarían reflejando el grado en que cada estudiante tiene asimilado el uso natural de las operaciones lógicas (el empleo de operadores de tipo booleanos por ejemplo) a situaciones problema, por ejemplo, como la aquí planteada respecto a la construcción de una molécula de ADN_r la cual requiere de la abstracción de ciertos conceptos. En el caso puntual de la abstracción, la misma ocupa el nivel más alto dentro del PC lo que implica que su desarrollo requiera del entrenamiento en el manejo de la conceptualización y del involucramiento de actividades cognitivas superiores [7]. En este sentido, el equipo de Scratch señala que las y los estudiantes al utilizar dicho lenguaje para programar aprenden a, como se indicó anteriormente, resolver problemas de manera colaborativa al tiempo que comienzan a desarrollarse sus habilidades computacionales.

En la instancia del post test, las habilidades del PC que se consideraron fueron las siguientes: secuenciación,

repetición (*looping*), control de flujo, paralelismo y sincronización y coordinación.

Para identificar el desarrollo de las mismas, se le propuso al grupo de estudiantes códigos programados según los bloques de Scratch que representaban a cada una de las habilidades mencionadas anteriormente. La repetición fue identificada correctamente por el 65% del estudiantado siendo la única habilidad que no había sido incluida hasta el momento lo que demostraría que las actividades propuestas y en el orden en el que fueron presentadas al estudiantado les permitió el desarrollo de esta habilidad. Con el 62% sigue la coordinación y sincronización, resultado respaldado por el análisis en [©]Dr. Scratch donde esta habilidad fue la más desarrollada por las y los estudiantes. De manera similar ocurrió con el paralelismo (59%). La secuenciación que en el pre test fue reconocida por el 97% de la cohorte, en este caso lo fue por el 44% lo que podría deberse a que los códigos para ejemplificar dicha habilidad no hayan sido los adecuados y que por tal motivos el estudiantado reconoció en menor medida a dicha habilidad en esa instancia. Finalmente, en el caso del control de flujo fue reconocida correctamente por el 38% de las y los estudiantes. En el análisis realizado por [©]Dr. Scratch es una habilidad que se encuentra en proceso de desarrollo lo que explicaría el resultado obtenido lo que implicaría que de haber contado con más tiempo para la implementación de la secuencia dicha habilidad podría haberse desarrollado en mayor medida.

Luego de finalizadas las actividades de la secuencia didáctica, se les consultó a las y los estudiantes en entrevistas a grupos focales sobre la incorporación de Scratch en sus clases. Las respuestas obtenidas coinciden con los resultados registrados por Iskrenovic-Momcilovic [33] quien encontró que estudiantes que participaron de actividades en Scratch pudieron crear su propio mundo creativo, que les resultó "genial" y divertido al tiempo que les brindaba oportunidades para la construcción de conocimientos de manera colaborativa con otros pares. En ambos casos se encontraron respuestas positivas respecto a la inclusión de Scratch en las clases. En línea con lo anterior, se les preguntó acerca del grado de dificultad de las actividades planteadas en la secuencia. Expresaron que al comienzo utilizar Scratch resultaba lo más difícil pero luego a medida que pasaban las instancias y comprendían su funcionamiento les resultaba fácil utilizarlo.

En el post test también se les consultó respecto a la importancia de trabajar en grupo durante el desarrollo del proyecto en Scratch sobre la insulina recombinante. El 74% de las respuestas se agruparon en relación a la colaboración y a la discusión de ideas para la elaboración del proyecto. En relación a seguir incursionando en el mundo de la programación, se encontraron respuestas comparables a las obtenidas Zhang y Nouri [5] quienes encontraron que a las y los estudiantes les gustaba la programación al tiempo que les resultaba entretenida y además indicaron que querían seguir aprendiendo sobre la programación lo que sostiene a Scratch factible de utilizar

como un primer acercamiento a la programación como lo fue en este grupo de estudiantes.

Posteriormente, en el post test se le preguntó al estudiantado con qué otros contenidos de Biología creían que les sería útil trabajar con Scratch. El 41% se inclinó por los sistemas del cuerpo humano, en cantidades iguales (15%) optaron por célula e hicieron referencia a que se podría trabajar con cualquier tema. Estos resultados sugerirían que Scratch es reconocido como una herramienta de aprendizaje factible de ser empleada en el desarrollo de diferentes contenidos en Biología y principalmente con aquellos que implican mayores niveles de abstracción como lo son estructura y reproducción celular o biomoléculas.

En este sentido, la utilización de Scratch como mediador de la modelización podría proponerse para otros contenidos dentro de la Biología como así también implementarse en diferentes diseños didácticos en otras Ciencias Naturales considerando que los resultados aquí expuestos podrían obtenerse en otros espacios curriculares.

Conclusiones

En esta investigación se desarrolló un estudio con el fin de caracterizar la integración de Scratch como herramienta mediadora de la modelización de la IG en la enseñanza de la Biología en la escuela secundaria.

Luego de participar de las actividades el estudiantado reconoció como uno de sus aprendizajes a Scratch lo que indica que utilizar este lenguaje como una primera aproximación a la programación permite que las y los estudiantes se familiaricen con los principales fundamentos de la programación. Además, al tratarse Scratch de un entorno que favorece la comunicación y el trabajo colaborativo promovió el uso de habilidades críticas, metacognitivas y reflexivas desde una perspectiva creativa y exploratoria para la resolución de problemas. En concordancia, numerosas investigaciones posicionan a Scratch como una herramienta con diversas potencialidades para la enseñanza y el aprendizaje de la programación [5]. Si bien algunos informáticos reconocen el valor de Scratch ya que no se necesita conocer todos los conceptos informáticos para comenzar a programar con dicho lenguaje, otros sostienen que algunas estructuras, rara vez se utilizan en los proyectos ya que no son fáciles de descubrir sin ayuda [26]. Sin embargo, la inclusión de Scratch permitió el desarrollo de algunas de las habilidades del PC.

En relación a las habilidades del PC puestas en juego por las y los estudiantes cuando modelizaron la construcción de una molécula de ADN en Scratch, se encontró un desarrollo diferencial de las habilidades del PC observándose un gradiente decreciente desde la sincronización, el paralelismo, la interactividad con el usuario, el control de flujo, la abstracción, la representación de la información y hasta el pensamiento

lógico. En línea con lo anterior, la incorporación de Scratch en las clases de ciencia puede contribuir al desarrollo de algunas habilidades de orden superior y directamente relacionada con la adquisición de competencias clave para el estudiantado. Entre dichas habilidades se encuentra la capacidad de plantear interrogantes con respecto a una situación determinada; de proponer hipótesis y modelos conceptuales para explicar la situación inicial; de producir o recopilar información con el propósito de verificar el modelo propuesto es decir, habilidades que utiliza la comunidad científica para la construcción de conocimiento.

Asimismo, las personas requieren de nuevas habilidades asociadas al PC para desenvolverse en los entornos digitales que cotidianamente impactan en su vida. Si bien a la fecha existen escasas investigaciones que permitan dar cuenta de un abordaje transversal y significativo de las ciencias de la computación, este trabajo constituye una aproximación a ello integrando a las ciencias de la computación con la Biología. Al respecto, Esteve-Mon y colaboradores [34] resaltan un cambio de paradigma conducente a que las habilidades en TIC sean consideradas transversales a otros espacios curriculares para promover el desarrollo de entornos de aprendizajes proponiendo nuevas metodologías de aprendizaje y de evaluación respecto al PC [1] estableciendo marcos conceptuales para la integración de las ciencias de la computación en los diseños curriculares de la escuela secundaria.

Las y los estudiantes encontrarán al finalizar sus estudios secundarios una sociedad donde unir el conocimiento humano a las tecnologías será crucial para su inserción en el mundo donde el PC ocupará un lugar preponderante siendo una competencia básica que la ciudadanía deberá conocer para desenvolverse en la sociedad digital permitiendo la resolución de problemas de manera colaborativa [13]. En este contexto los saberes digitales se constituyen en un derecho que debe fomentarse y protegerse para desarrollar capacidades para interactuar en un contexto digital flexible y variable [4].

Los conocimientos presentados en este artículo ejemplifican una manera en la cual se pueden integrar el PC y las prácticas científicas en la enseñanza de la Biología promoviendo la utilización de recursos computacionales para expresarse, resolver problemas y modelizar, esta última una práctica científica que presenta muchas potencialidades dentro de la enseñanza de las ciencias.

Se requieren investigaciones que permitan determinar estructuras, componentes y metodologías que resulten en la definición de criterios para la integración del PC en la escuela secundaria y más específicamente en la enseñanza de las ciencias contribuyendo a la formación de personas garantizando el desarrollo de sus saberes, digitales y científicos, para que puedan desenvolverse plena y sustentablemente en el mundo.

Agradecimientos

Grupo de Investigación EDUCEVA-CienciaTIC. Secretaría de Ciencia y Tecnología (SECyT). Universidad Nacional de Córdoba. Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).

Referencias

- [1] E. Coronel Díaz and G. Lima Silvain, "El pensamiento computacional. Nuevos retos para la educación del siglo XXI," *Virtualidad, Educación y Ciencia*, vol. 20, no. 11, pp. 115-137, 2020.
- [2] M. B. Bonello and F. Schapachnik, "Diez preguntas frecuentes (y urgentes) sobre pensamiento computacional," *Virtualidad, Educación y Ciencia*, vol. 20, no. 11, pp. 156-167, 2020.
- [3] G. Puttick and E. Tucker-Raymond, "Building Systems from Scratch: An Exploratory Study of Students Learning About Climate Change," *Journal of Science Education and Technology*, vol. 27, pp. 306-321, 2018.
- [4] J. Valverde Berrocoso, M. R. Fernández Sánchez and M. C. Garrido Arroyo, "El pensamiento computacional y las nuevas ecologías del aprendizaje," *RED. Revista de Educación a Distancia*, no. 46, 2015, Art. 3.
- [5] L. Zhang and J. Nouri, "A systematic review of learning computational thinking through Scratch in K-9," *Computers and Education*, vol. 141, 2019.
- [6] K. Brennan and M. Resnick, "Imagining, Creating, Playing, Sharing, Reflecting: How Online Community Supports Young People as Designers of Interactive Media," in *Emerging Technologies for the Classroom, Explorations in the Learning Sciences, Instructional Systems and Performance Technologies*, C. Mouza and N. Lavigne, Eds., Nueva York: Springer, 2013, pp. 253-268.
- [7] J. M. Wing, "Computational thinking and thinking about computing," *Philosophical Transactions of the Royal Society a Mathematical Physical and Engineering Sciences*, vol. 366, no. 1882, pp. 3717-3725, 2008.
- [8] B. France, "Socially Acute Questions: How Biotechnology Can Provide Context and Content for Discussion in Food Technology Education," in *Food Education and Food Technology in School Curricula. Contemporary Issues in Technology Education*, M. Rutland and A. Turner, Eds., Cham: Springer, 2020, pp. 273-282.
- [9] M. Resnick and N. Rusk, "Coding at a Crossroads," *Communications of ACM*, vol. 63, no. 11, pp. 120-127, 2020.
- [10] M. Román-González, J. C. Pérez-González and C. Jiménez-Fernández, "Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test," *Computers in Human Behavior*, vol. 72, pp. 678-691, 2016.
- [11] National Research Council, *Committee for the Workshops on Computational Thinking: Report of a workshop on the scope and nature of computational thinking*. Washington, DC, U.S.: National Academies Press, 2010.
- [12] S. Bocconi, A. Chiocciariello, G. Dettori, A. Ferrari and K. Engelhardt, *Developing computational thinking in compulsory education. Implications for policy and practice*. Luxembourg: Publications Office of the European Union, 2016.
- [13] J. M. Wing, "Computational thinking," *Communications of the ACM*, vol. 49, no. 3, pp. 33-35, 2006.
- [14] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman and Y. B. Kafai, "Scratch: Programming for all," *Communications of the ACM*, vol. 52, no. 11, pp. 60-67, 2009.
- [15] D. A. Fields, Y. B. Kafai and M. T. Giang, "Youth Computational Participation in the Wild: Understanding Experience and Enquiry in Participating and Programming in the Online Scratch Community," *AMC Transactions on Computing Education*, vol. 17, no. 3, pp. 1-22, 2017.
- [16] International Society for Technology in Education (ISTE) y The Computer Science Teachers Association (CSTA), *Operational definition of computational thinking for K-12*, 2011.
- [17] J. M. Wing, "Research notebook: Computational thinking. What and why?," *The Link Magazine*, Spring, pp. 20-23. Carnegie Mellon University, Pittsburgh. [Online]. Available: <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>
- [18] The Royal Society, *Shut down or restart? The way forward for computing in UK schools*. London: The Royal Society Education Section, 2012.
- [19] J. Moreno-León, G. Robles and M. Román-González, "Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking," *RED: Revista de Educación a Distancia*, no. 46, 2015, Art. 10.
- [20] H. U. Hoppe and S. Werneburg, "Computational Thinking. More Than a variant of Scientific Inquiry!," in *Computational Thinking Education*, S. Kong and H. Abelson, Eds., Singapur: Springer, 2019, pp. 13-30.
- [21] A. Yadav, S. Gretter, J. Good and T. McLean, "Computational Thinking in Teacher Education," in *Emerging Research, Practice and Policy on*

- Computational Thinking. Educational Communications and Technology: Issues and Innovations*, P. Rich and C. Hodges, Eds., Cham: Springer, 2017, pp. 205-220.
- [22] O. Meerbaum-Salant, M. Armoni and M. Ben-Ari, "Learning computer science-concepts with Scratch," *Computer Science Education*, vol. 23, no. 3, pp. 239-264, 2013.
- [23] O. Erol and A. A. Kurt, "The effects of teaching programming with Scratch on pre-service information technology teacher's motivation and achievement," *Computers in Human Behavior*, vol. 77, pp. 11-18, 2017.
- [24] N. Calder, "Using Scratch to facilitate mathematical thinking," *Journal Education*, vol. 23, no. 2, pp. 43-48, 2018.
- [25] Y. B. Kafai, D. A. Fields, R. Roque, W. Q. Burke and A. Monroy-Hernández, "Collaborative agency in youth online and offline creative production in Scratch," *Research and Practice in Technology Enhanced Learning*, vol. 7, no. 2, pp. 63-87, 2015.
- [26] M. Guzdial, "What we care about now, what we'll care about in the future," *ACM Inroads*, vol. 9, no. 4, pp. 63-64, 2018.
- [27] R. M. Kelly, "Aprender de las animaciones de contraste molecular con una actividad de monitorización metacognitiva," *Educación Química*, vol. 20, no. 23, pp. 181-194, 2017.
- [28] M. Haskel-Ittah and A. Yarden, "Toward Bridging the Mechanistic Gap Between Genes and Emphasizing the Role of Proteins in a Computational Environment," *Science and Education*, vol. 26, no. 10, pp. 1143-1160, 2017.
- [29] S. Veselinovska and A. Stavreva, "The impact of the usage of web animation in teaching molecular and cellular biology," *Journal of Educational Science. Theory and Practice*, vol. 11, no. 15, pp. 116-130, 2020.
- [30] J. K. Gilbert and R. Justi, "Introducing Modelling into School Science," in *Science Education Research and Practice in Asia-Pacific and Beyond*, J. Yeo, T. Teo and K. S. Tang, Eds., Singapur: Springer, 2018, pp. 25-38.
- [31] P. Cobb, K. Jackson and C. Dunlap, "Design research: An analysis and critique," in *Handbook of International Research in Mathematics Education*, L. D. English and D. Kirshner, Eds., Nueva York: Routledge, 2016, pp. 481-503.
- [32] E. E. Scott, M. P. Wenderoth and J. H. Doherty, "Design-Based Research: A Methodology to Extend and Enrich Biology Education Research," *Life Sciences Education*, vol. 19, no. 3, 2020.
- [33] O. Iskrenovic-Momcilovic, "Pair programming with Scratch," *Education and Information Technologies*, vol. 25, pp. 2943-2952, 2019.
- [34] F. Esteve-Mon, M. Llopis and J. Adell-Segura, "Digital Competence and Computational Thinking of Student Teachers," *International Journal of Emerging Technologies in Learning*, vol. 15, no. 2, pp. 29-41, 2020.

Información de Contacto de las Autoras:

Gimena Betina Fussero

Av. Vélez Sarsfield 299
Córdoba
Argentina

gimenafussero@unc.edu.ar

www.inv.educeva.efn.uncor.edu

ORCID ID: <https://orcid.org/0000-0003-0213-0637>

Maricel Ocellli

Av. Vélez Sarsfield 299
Córdoba.
Argentina

maricel.occelli@unc.edu.ar

www.inv.educeva.efn.uncor.edu

ORCID ID: <https://orcid.org/0000-0002-4516-0644>

Marcela Chiarani

Av. Ejército de los Andes 950
San Luis
Argentina

mcchiarani@gmail.com

ORCID ID: <https://orcid.org/0000-0001-8440-4456>

Gimena Betina Fussero

Mg. en Educación en Ciencias Experimentales y Tecnología; Bióloga y Profesora en Cs. Biológicas. FCEFYN. Universidad Nacional de Córdoba. Grupo EDUCEVA-CienciaTIC. Prof. en nivel medio y superior. Prof. Asistente.

Maricel Ocellli

Dra. en Ciencias de la Educación. Mg. en Educación en Ciencias Experimentales y Tecnología. Bióloga. Prof. en Cs. Biológicas. FCEFYN. Universidad Nacional de Córdoba. Grupo EDUCEVA-CienciaTIC. Investigadora CONICET. Prof. Titular.

Marcela Chiarani

Mg. en Tecnología de la Educación. Licenciada en Ciencias de la Computación. Directora del Centro de Informática Educativa. Departamento de Informática. FCFMyN. Universidad Nacional de San Luis. Prof. Asociada.