e-Book | Actas

Actas II Jornadas Argentinas de Didáctica de la Programación

Editores:

Araceli Acosta
Belén Bonello
Cecilia Martínez
Sonia Permigiani
Nicolás Wolovick

UNC | FAMAF | ffyh | UNRC

ACTAS

II JORNADAS ARGENTINAS DE DIDÁCTICA DE LA PROGRAMACIÓN

Actas II Jornadas Argentinas de Didáctica de la Programación / Alejandro Iglesias... [et al.] ; editado por Araceli Acosta... [et al.].- 1a ed.- Córdoba : Universidad Nacional de Córdoba. Facultad de Filosofía y Humanidades, 2020.
Libro digital, PDF

Archivo Digital: descarga y online
ISBN 978-950-33-1600-9

1. Didáctica. 2. Lenguaje de Programación. 3. Formación Docente. I. Iglesias, Alejandro. II. Acosta, Araceli, ed.
CDD 004.071

COMITÉ ACADÉMICO

Araceli Acosta
Marcelo Arroyo
Francisco Bavera
Luciana Benotti
María Belén Bonello
Virginia Brassesco
Claudia Casariego
Marcela Daniele
Gladys Dapozo
Gustavo Del Dago
Maria Emilia Echeveste
Marcos Gomez
Carolina Gonzalez
Guillermo Grosso
Renata Guatelli
Marta Lasso
Maria Carmen Leonardi
Diego Letzen
Matías López Rosenfeld
Cecilia Martinez
Pablo E. Martínez López
Analia Mendez
Natalia Monjelat
Sonia Permigiani
María Valeria Poliche
Claudia Queiruga
Jorge Rodríguez
Alvaro Ruiz-de-Mendarozqueta
Claudia Cecilia Russo
Alfredo Héctor Sanzo
Fernando Schapachnik
Herman Schinca
Pablo Turjanski
Nicolás Wolovick
Dante Zanarini
Rafael Zurita

EDITORES

Araceli Acosta
Belén Bonello
Cecila Martínez
Sonia Permigiani
Nicolás Wolovick

ILUSTRACIÓN DE TAPA

Manuel Coll – Área de Comunicación Institucional – FFyH – UNC

Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional.

Índice

Introducción 6

SECCIÓN N°1 | CURRÍCULUM Y MATERIALES DE ENSEÑANZA

Colección de actividades desconectadas para el desarrollo de pensamiento computacional en el nivel primario 12
Alejandro Iglesias; Fernando Bordignon

Potencialidad didáctica de las actividades experimentales abiertas para la enseñanza de disciplinas STEM 23
Víctor Furci; Oscar Trinidad; Fernando Bordignon; Luis Peretti

Robótica educativa con software libre y hardware de especificaciones abiertas para enseñanza de programación 47
Diego Letzen; Valentín Basel; Alba Massolo; Federico Ferrero

Acercando la programación a la escuela secundaria con RITA 57
Isabel Miyuki Kimura; Vanessa Aybar Rosales; Claudia Queiruga

Wollok: Un entorno de aprendizaje de Programación Orientada a Objetos 64
Lucas Spigariol

Estrategias de enseñanza de la programación y su aplicación en aulas con estudiantes con discapacidad 79
María Fernanda Golobisky, Rosana Portillo

SECCIÓN N°2 | FORMACIÓN DOCENTE

Hacia una didáctica de la programación para la secundaria argentina 88
Pablo E. "Fidel" Martínez López, Alfredo Sanzo, Fernando Schapachnik

Formación docente y Currículum en el campo de las Ciencias de la Computación en dos provincias Argentinas 100
María Cecilia Martínez; Natalia Monjelat

Enseñar a Enseñar Ciencias de la Computación. Una experiencia sobre políticas educativas y contenidos de Ciencias de la Computación 109
Claudia Queiruga; Claudia Banchoff Tzancoff; Soledad Gómez; Paula Venosa

Análisis de producciones de docentes de Educación primaria en formación sobre Didáctica de las Ciencias de la Computación 121
Marcela Daniele; Teresa Quintero; Cecilia De Dominici; Flavia Buffarini; Francisco Bavera

La enseñanza de la programación a través del ABP. Una experiencia de formación en el Profesorado de Informática 132
Silvia Pilar Rodríguez

Introducción

En junio del año 2019 se llevaron a cabo las segundas Jornadas de Didáctica de la Programación en la Universidad Nacional de Córdoba. Nuestros colegas y amigos de la Universidad Nacional de Quilmes y de la Fundación Sadosky (Pablo "Fidel" Martinez López, Mara Bordchart, Belén Bonello y Fernando Schapachnick) tuvieron el impulso de organizar las primeras Jornadas en 2018 para consolidar y seguir promoviendo una comunidad educativa que aborda temas de enseñanza de las Ciencias de la Computación. Nos dieron el pase, como hacen los buenos jugadores que se sienten parte de un equipo. Así, entre las Universidades Nacionales de Córdoba y Río Cuarto conformamos un nuevo comité para las JADiPro 2019[1].

En los objetivos principales del encuentro, se planteaba la necesidad de reunir y escuchar a la comunidad educativa de todo el país que estaba realizando innovaciones que promueven la inclusión de las Ciencias de la Computación en la escuela. Por ello, en la edición de 2019 se propusieron encuentros con seis formatos diferentes para ampliar la participación de nuestra comunidad.

Estamos en un momento de profundos cambios en el campo de la enseñanza de la informática como resultado de un trabajo sostenido que comenzó en 2012 con Program.AR, programa de la Fundación Sadosky. En 2015 el Consejo Federal de Educación resuelve que la enseñanza de la programación es estratégica, y en el 2018 incluye contenidos de programación y robótica en los Núcleos de Aprendizaje Prioritarios (NAPs) que abarcan a todo el territorio nacional. Como consecuencia, todas las provincias del territorio deben incluir estos contenidos en sus bases curriculares y diseñar estrategias para garantizar su enseñanza.

Más allá de las normativas y políticas educativas, como educadores estamos comprendiendo cada vez más que para ejercer una ciudadanía de pleno derecho es necesario entender el lenguaje digital y los conceptos centrales a partir de los cuales se organizan plataformas, algoritmos, dispositivos, comunicaciones, etc, que atraviesan intensivamente nuestro quehacer cotidiano. Al decir de Amy Ko y sus colegas (2020)

> "…..*La creciente demanda de información está aumentando la explotación minera y la producción de carbono. Las redes sociales amplifican falsedades einternet es el nuevo campo de batalla en la guerra moderna. Y en todos estos sistemas, los datos y algoritmos amplifican el racismo, el sexismo, el heterosexismo, el capacitismo, la discriminación por edad, la xenofobia, la cisheteronormatividad y otras formas de*

1 Las JADiPro fueron avaladas por la Unión de Educadores de la Provincia Córdoba (UEPC) y el Córdoba Technology Cluster. Asimismo nos apoyaron el Ministerio de Ciencia de la Provincia de Córdoba, la Agencia Nacional de Promoción Científica y Tecnológica, el Grupo de Interés especial en enseñanza de las Ciencias de la Computación de la ACM, la Agencia de Córdoba Turismo y la Fundación Sadosky.

inequidad, injusticia y prejuicio...Estos vínculos entre la computación y la injusticia parecen invisibles para muchos, incluso para quienes resultan los más afectados. Muchos jóvenes crecen viendo a las computadoras como máquinas mágicas que brindan alegría, escape y conexión; otros las experimentan como vectores de violencia, acoso sexual, ciberacoso, adicción y aislamiento….Argumentamos, como lo han hecho otros, que hacer visibles estas injusticias a la sociedad es la responsabilidad de los educadores de Ciencias de la Computación". (Ko, Oleson, Ryan, Register, Xie, Tari, Davidson, Druga, Loksa, 2020)

Porque incluye a todos y a todas, porque su rol es la democratización de los saberes que como sociedad hemos construido para que esté al alcance de cada uno, porque es el lugar para aprender algo diferente a lo que ofrece el núcleo de socialización primaria, porque es una institución diseñada para aprender con otros distintos pero de una misma generación, y porque es una de las pocas instituciones que ha seguido funcionando cuando el mundo se paró en 2020; es que la escuela es el lugar para ofrecer saberes imprescindibles como lo son los del campo de las Ciencias de la Computación. Y es por eso que muchas provincias argentinas han comenzado a formar a docentes para que aborden este campo del conocimiento en sus aulas y han incluido contenidos de Ciencias de la Computación en sus bases curriculares. Nuestra apuesta en estas jornadas era conocer y aprender de muchos de ellos. ¡Y lo logramos!

Reunimos a más de 400 docentes de todos los niveles educativos: desde nivel inicial hasta universitario. Convocamos a colegas de Uruguay y Chile para que nos contaran sus programas nacionales. Se sumaron educadores de 17 provincias Argentinas: Tucumán, Santiago del Estero, Santa Fe, Santa Cruz, San Luis, Salta, Neuquén, Misiones, Mendoza, La Rioja, La Pampa, Entre Ríos, Corrientes, Chaco, Catamarca, Córdoba y Buenos Aires. Todas ellas y ellos participaron en alguno de los formatos propuestos:

1. Presentación de Posters. Se presentaron 13 posters que están disponibles en la página del evento: https://jadipro.unc.edu.ar/trabajos-presentados/

2. Simposios: en una apuesta de promover la articulación interinstitucional, las y los colegas se organizaron en una presentación para analizar temas emergentes recuperando las miradas de cada uno. En total se organizaron cuatro simposios que nombramos a continuación:

- **Políticas de Formación Docente.** Gonzalo Gutierrez (UEPC), Mara Bordchard y Javier Castrillo (Fundación Sadosky), Ricardo Medel (ISEP).
- **La Brecha Digital de Género en distintos ámbitos educativos.** Soledad Salas (Córdoba Technology Cluster), Emilia Echeveste y Marcos Gómez (UNC-CONICET), Laura Alonso (FAMAF) e Ivana Feldfeber (Mumuki).
- **¿Cómo combinar los enfoques unplugged y plugged en la formación de docentes del nivel primario? Distintas aproximaciones y experiencias.** Ana

Casali y Natalia Monjelat (UNR-CIFASIS e IRICE), María Carmen Leonardi y María Virginia Mauco (UNCPBA), Francisco Bavera y Flavia Buffarini (UNRC).

- **Robótica Educativa. ¿Piso Bajo? o ¿Techo Bajo?** Valentín Basel (CIECS), Eduardo Rodríguez y Martín Onetti (CREC, FAMAF), Edgardo Silvi (MinCyT), Guillermo Grosso, (UNCOMA).

3. Talleres: para ofrecer un espacio de formación de pares, colegas de diferentes provincias diseñaron y postularon una propuesta de formación para docentes. Se seleccionaron cuatro talleres en los que participaron más de 300 docentes recibiendo puntaje oficial del Ministerio de Educación de la Provincia de Córdoba. Los nombramos a continuación:

- **Programación y Robótica Educativa.** A cargo de Luciana Cardozo, profesora de la Escuela Técnica de Misiones.
- **Resolución de Problemas que favorecen Procesos de Generalización: Un Análisis Didáctico-Matemático Como Marco de Reflexión.** A cargo de Flavia Buffarini, Fabiana Rosso, Francisco Bavera y Marcela Danieley su equipo de la Universidad Nacional de Río Cuarto.
- **Computadoras en Papel**. A cargo de Valentín Basel (CONICET) y Gustavo del Dago, (Fundación Sadosky, Prof. de Computación en Escuelas Técnicas), Nicolás Wolovick (UNC).
- **¿Cómo enseñarle a programar a nuestros jóvenes a través de la indagación?** A cargo de Pablo Fidel Martinez Lopez de la Universidad Nacional de Quilmes.

4. Reunión de doctorandos: se habilitó un espacio de intercambio de avances entre doctorandos y profesores de todo el país que son pioneros en realizar sus tesis en temas de educación en informática. En total 10 doctorandos de todo el país pudieron presentar sus avances y recibir comentarios de pares y docentes.

5. Paneles: los organizadores del evento armamos 3 paneles con invitados especiales del campo de la comunicación, divulgación científica, comunidad de software libre, y referentes de programas y políticas educativas.

6. Mesas de trabajo: 5 mesas de trabajo de un promedio de 4 participantes cada una presentaron artículos de investigación o reflexión didáctica. Estas actas presentan las versiones corregidas de los artículos seleccionados.

Los 11 artículos se organizaron en dos grandes secciones: 1) Currículum y Materiales de Enseñanza y 2) Formación Docente. Si bien la convocatoria incluía otros ejes de análisis, los trabajos de las y los colegas abordaban solo estos dos.

En la primera sección “Currículum y Materiales de Enseñanza” Alejandro Iglesias y Fernando Bordignon nos invitan a pensar sobre qué contenidos y operaciones cognitivas

permiten diferentes tipos de actividades para enseñar a programar. Los autores recuperan las ya clásicas actividades unplugged y de programación de bloques entre otras para analizarlas como dispositivo didáctico: qué permiten aprender y desarrollar cada una de ellas. A partir de este análisis nos ofrecen un marco de referencia para pensar sobre los materiales que elaboramos en este momento tan prolifero de desarrollo didáctico en el área.

Del mismo modo, Diego Letzen, Valentín Basel, Alba Massolo y Federico Ferrero recuperan múltiples recursos para enseñar computación a través de la robótica y analizan las potencialidades didácticas y limitaciones de cada uno. En conjunto, estos autores nos recuerdan que elegir un material de enseñanza supone tener conciencia y claridad sobre qué contenidos hemos seleccionado para transmitir a nuestros estudiantes y para qué decidimos enseñar estos conceptos.

Víctor Furci, Oscar Trinidad, Fernando Bordignon y Luis Peretti indagan sobre la posibilidad de integrar la enseñanza de las Ciencias de la Computación en proyectos STEM. La organización de los contenidos de Ciencias de la Computación es un tema no resuelto desde las políticas educativas y tampoco en las comunidades escolares. Integrado, transversal o con espacio curricular propio es un debate vigente y requiere de un profundo análisis. El estudio de los colegas de UNIPE permite analizar críticamente propuestas didácticas integradas relacionando las actividades propuestas con las disposiciones cognitivas que activa cada una. Recuperan también la esencial mirada de los docentes en el proceso de desarrollo de materiales educativos.

En esa misma línea los artículos de Isabel Miyuki y Lucas Spigarol analizan dos entornos didácticos desarrollados en sus equipos de trabajo para aprender a programar: RITA y WOLLO, respectivamente. En ambos casos describen con detalle los aspectos técnicos y didácticos. Recuperan los comentarios de docentes que han usado las herramientas.

Finalmente, María Fernanda Golobisky y Rosana Portillo llevan a cabo una experiencia única enseñando nociones básicas de programación a niños y niñas con discapacidad. A partir de los resultados donde evidencian apropiaciones conceptuales, sugieren y nos invitan a pensar en el potencial de la enseñanza de la programación para el desarrollo cognitivo.

En la segunda sección “Formación Docente”, Cecilia Martinez y Natalia Monjelat documentan los dos primeros postítulos del país en Especialización Docente en Enseñanza de la Programación en Córdoba y Santa Fe. El trabajo describe el estado de la formación docente en computación en ambas provincias y la ruptura del paradigma de enseñanza de la computación que supone la oferta curricular del postítulo.

El artículo de Pablo “Fidel” Martinez Lopez describe con detalle los criterios pedagógicos que se construyeron para seleccionar contenidos y estrategias de formación docente en conceptos centrales de la programación en un curso de formación docente. Todavía es un debate qué contenidos seleccionamos dentro del campo de la informática y cómo los acercamos a la formación docente. Esta reflexión hace un aporte en esta dirección.

Como parte de estos esfuerzos de formación docente se encuentran los postítulos de la Universidad Nacional de La Plata y de la Universidad Nacional de Río Cuarto. Claudia Queiruga, Claudia Banchoff Tzancoff, Soledad Gómez y Paula Venosa de la Universidad de la Plata describen y analizan el diseño de contenidos de su propuesta de formación docente y presentan resultados sobre las percepciones de los maestros. La relevancia de poder documentar estas experiencias radica en que son escasos los programas de formación docente inicial o especializaciones existentes en el área, y sabemos muy poco sobre con qué criterios seleccionar contenidos disciplinares y pedagógicos para armar estos currículums de formación docente. En ese sentido, la documentación de estas experiencias es un aporte necesario.

Por su parte, los colegas de Río Cuarto, Marcela Daniele, Teresa Quintero, Cecilia De Dominici, Flavia Buffarini y Francisco Bavera nos muestran un análisis detallado de los aprendizajes de los y las docentes a partir de indagar sobre sus producciones. En la lectura analítica de estos proyectos encuentran que las propuestas reflejaron habilidades cognitivas relacionadas al mundo digital, mostrando diferentes niveles de desarrollo y profundidad. Desde este trabajo exploratorio e inicial, se intenta aportar al desarrollo de conocimientos sobre la formación continua de docentes de Educación Primaria en Didáctica de las Ciencias de la Computación y el impacto en sus aulas de la escuela primaria.

Finalmente el estudio de Pilar Rodriguez analiza un enfoque didáctico para abordar la enseñanza de la computación entre los docentes: el Aprendizaje Basado en Proyectos (ABP). Con detalle descriptivo logra comunicar los criterios y potencialidades de aprendizaje que ofrece este formato en particular para el caso de la introducción de la programación -que no tiene un lugar específico en el currículum de los profesores en formación.

En conjunto los trabajos muestran a una comunidad en movimiento, centrada en generar innovaciones educativas desde dos instrumentos centrales de las políticas educativas: la formación docente y el currículum con sus materiales. El 2020 obligó a todos y todas las docentes a capitalizar en estos desarrollos y en re contextualizar los materiales para la enseñanza remota.

Los trabajos presentados también muestran la necesidad de documentar cómo esas innovaciones son recibidas en las escuelas por docentes y, lo más importante, por estudiantes, más allá del contacto directo con las universidades que las promueven. Queda mucho por seguir preguntándonos y reflexionando como comunidad. ¿Cómo aprenden los diferentes estudiantes? ¿Qué relación encontramos entre la brecha digital y la oferta curricular en las escuelas? ¿Cuál es el estado de situación de los currículos actuales después de estos años? ¿Qué contenidos se jerarquizan y cuáles se dejan de lado en la enseñanza de la programación?, etc. Como buen equipo, seguimos pasando la pelota para las próximas JADiPro.

SECCIÓN N°1

CURRÍCULUM Y MATERIALES DE ENSEÑANZA

Colección de actividades desconectadas para el desarrollo de pensamiento computacional en el nivel primario

Alejandro Iglesias[1], Fernando Bordignon[2]

Resumen: En el año 2018 el Consejo Federal de Educación definió nuevos Núcleos de Aprendizaje Prioritarios (NAP) que incluyen contenidos relacionados con la programación y la robótica. En particular, no solo se introdujeron conceptos básicos de programación de computadoras, sino que también se indicó la necesidad del desarrollo de capacidades en torno la resolución de problemas utilizando tecnología digital. Por este motivo, entendemos que se deben implementar experiencias didácticas que permitan, además de aprender a programar, el desarrollo del pensamiento computacional (Wing, 2006) en los estudiantes. Las actividades desconectadas constituyen un primer acercamiento al desarrollo del pensamiento computacional que, potencialmente, pueden ser trabajadas en todos los establecimientos educativos ya que no se requieren de una infraestructura tecnológica particular. Desde la Universidad Pedagógica Nacional se están desarrollando recursos didácticos centrados en este tipo de actividades. En este trabajo presentamos una colección de actividades que se centran en la comprensión de conceptos por parte del maestro así como de los estudiantes. La colección se enfoca en el desarrollo de capacidades del pensamiento computacional de manera incremental, partiendo de ejercicios simples y avanzando hacia el desarrollo de la creatividad a partir de trabajar con problemas.

Palabras claves: pensamiento computacional, actividades desconectadas, segundo ciclo de escuela primaria argentina.

1 Universidad Pedagógica Nacional, alejandro.iglesias@unipe.edu.ar
2 Universidad Pedagógica Nacional, fernando.bordignon@unipe.edu.ar

Introducción

En el año 2018 el Consejo Federal de Educación ha definido nuevos Núcleos de Aprendizaje Prioritarios (NAP) que están en relación con la programación y la robótica (Resolución Consejo Federal de Educación N° 343/18). Entre estos saberes nuevos, a incluir en los niveles inicial, primario y secundario, además de los temas relacionados con el uso de herramientas y con la ciudadanía digital, se incluyen temas que promueven el desarrollo de capacidades en torno a la resolución de problemas, la programación y la creación de aplicaciones. En este contexto, el concepto de pensamiento computacional (PC) se vuelve relevante para las escuelas argentinas.

El término "pensamiento computacional" es utilizado para hacer referencia a técnicas y metodologías de resolución de problemas, donde principalmente intervienen saberes que provienen de las ciencias de la computación. El desarrollo del PC está más allá de la formación de técnicos y profesionales informáticos. Según Jeannette Wing, promotora del concepto, el PC es una habilidad fundamental que debería ser desarrollada por todas las personas y no solo ser exclusiva de los profesionales del área (Wing 2006).

Las capacidades que están vinculadas al pensamiento computacional, pueden ser resumidas en cinco elementos claves (CAS 2015): a) el pensamiento algorítmico, referido a la capacidad de poder expresar soluciones a problemas a partir de una serie de pasos que un autómata puede llevar a cabo; b) la descomposición, referido a la capacidad de poder dividir e identificar las partes que componen un problema para facilitar su tratamiento y análisis; c) la generalización, entendida como la capacidad para descubrir patrones en los problemas o en las soluciones que son aplicables a ellos; d) la abstracción, que se refiere a la capacidad de poder elegir las representaciones que destacan las características relevantes a un contexto y ocultan los detalles innecesarios a él; y finalmente e) la evaluación, que se concibe como la capacidad de poder analizar críticamente las soluciones creadas para detectar y corregir posibles errores, así también como la búsqueda de soluciones que aprovechen mejor los recursos.

Existe una importante variedad de enfoques y estrategias didácticas destinadas a realizar prácticas escolares orientadas al desarrollo del PC (Kotsopoulos y otros, 2017). En particular, en el nivel primario, las propuestas tradicionales se enfocan en realizar actividades de programación utilizando plataformas como "Scratch" o bien utilizan plataformas en línea con entornos visuales que simplifican la interfaz de programación, a través del uso de "bloques encastrables" que se pueden arrastrar y soltar (McNerney 2004). En el plano internacional, la organización sin fines de lucro Code.org[3] ha desarrollado numerosas actividades que utilizan esta última forma de trabajo. Estas propuestas han resultado ser un enfoque interesante para los niños y jóvenes ya que se ha utilizado por personas particulares y por escuelas en decenas de países (Kumar 2014). En el ámbito local la Fundación Sadosky[4] ofrece también propuestas interesantes en este sentido.

3 http://code.org

4 http://www.fundacionsadosky.org.ar/

Otro enfoque complementario que puede utilizarse, especialmente en los primeros años de primaria, consiste en las denominadas “actividades desconectadas” o bien, en inglés, “unplugged” (Bell y Vahenrenhold, 2018). Estas actividades comprenden una gran variedad de ejercicios, juegos y problemas que se desarrollan sin requerir del uso de computadoras. Así, por ejemplo, existen actividades que desarrollan la capacidad de abstracción, otras que se basan en el reconocimiento de patrones y otras que pueden enfocarse en la explicación de cómo se crea o funciona un algoritmo en particular (Ozcinar y otros, 2017). En general, este tipo de estrategias didácticas abordan de manera separada los temas y las capacidades que componen el pensamiento computacional, aunque existen también propuestas, en minoría, que lo trabajan de forma integral.

Ambos enfoques (el conectado y el desconectado) se configuran como estrategias que pueden funcionar mejor si se usan una en complemento de la otra. Kostopoulos y su grupo de trabajo (Kostopoulos y otros, 2017) han propuesto un marco de trabajo (Computational Thinking Pedagogical. Framework, CTPF) para la enseñanza y el desarrollo del pensamiento computacional en los estudiantes. Propone utilizar de forma alternada, en una estructura de espiral, actividades desconectadas para desarrollar conceptos iniciales, avanzar con actividades de experiencias lúdicas (*tinkering*), luego realizar actividades relacionadas con el hacer y construir cosas y al final pasar a una etapa de remezcla, donde se modifican objetos creados por otras personas para darles nuevos usos. Este camino puede realizarse varias veces escalando en complejidad las habilidades, conceptos y tecnologías que se quieran trabajar. Es un enfoque integrado que se apoya en las ideas pedagógicas de Seymour Papert y Lev Vygotsky principalmente, donde el hacer con otros es el motor que permite construir aprendizajes significativos.

Actividades desconectadas

Las actividades desconectadas han tomado relevancia en la enseñanza del pensamiento computacional gracias a la iniciativa “Computer Science Unplugged” (CS Unplugged[5]) creada por la Universidad de Canterbury en Nueva Zelanda. El proyecto reúne una importante cantidad de problemas, clasificados por edad y nivel de complejidad, que tienen por objetivo desarrollar el PC en los niños y jóvenes. Empresas tales como Google y Microsoft apoyan y colaboran con la iniciativa. Así mismo, las actividades han sido traducidas a más de 20 idiomas (Bell, Vahrenhold y Fellow, 2008) y son utilizadas en más 50 países.

Se puede considerar que las actividades de tipo desconectadas comparten una serie de características (Bell-Vahenrenhold, 2018) (Tomohiro y otros, 2009) a saber:

- No utilizan computadoras.
- Tienen sentido lúdico.
- Presentan desafíos al estudiante para que éste explore distintos aspectos.

5 https://csunplugged.org/es/

- Suelen incorporar elementos de trabajo manual o corporal.
- Tienen un enfoque constructivista.
- Son sencillas y favorece el explicar cómo funcionan.
- Generalmente están acompañadas de elementos de fantasía que le dan cohesión y ayudan a desarrollar distintas metáforas que conectan con los mundos de los niños y de los jóvenes.

La naturaleza de estas actividades les confiere una serie de ventajas por sobre los acercamientos tradicionales en la enseñanza de la resolución de problemas ya que, por un lado, permite que el trabajo se centre en los problemas y los conceptos a trabajar y no en la tecnología asociada para resolverlos (no existen dificultades técnicas vinculadas, ni problemas derivados de las interfaces); por otro lado, permiten trabajar conceptos del PC con un menor nivel de abstracción, ya que en general se representan a través de metáforas o de objetos tangibles; y finalmente, no requieren de una infraestructura tecnológica especial para llevarse a cabo y por lo tanto tienen un bajo costo económico y son aplicables en casi cualquier institución.

Existe una variedad de actividades desconectadas, que a pesar de compartir criterios de diseño, son diferentes entre sí por su naturaleza. Esto provoca que existan propuestas que pueden resultar más o menos pertinentes para distintos grupos de estudiantes dependiendo de la edad, conocimientos previos y expectativas de los mismos (Taub y otros 2012). Es en este contexto, creemos que es necesario presentar una taxonomía, de elaboración propia, para ayudar a identificar y diferenciar los distintos tipos de actividades desconectadas. El criterio de clasificación está basado en las acciones que debe realizar el estudiante para la resolución. Algunas actividades pueden pertenecer a más de una categoría debido a que la clase solo indica el enfoque central. En concreto, se propone la siguiente clasificación:

Tabla I. Clasificación de actividades desconectadas.

Actividades kinestésicas
Son aquellas actividades que involucran movimientos físicos por parte de los estudiantes. Suelen ser apropiadas para el nivel inicial y los primeros años de la escuela primaria, sin embargo para los años o niveles superiores no resultan apropiadas, ya que no resultan atractivas y pueden llegar a desmotivar a los estudiantes (Taub y otros, 2012). Un ejemplo de este tipo de actividades podría ser construir con tarjetas una actividad que se base en un conjunto de instrucciones (podrían ser: “aplaudir” “saltar” “girar” en función de diseñar bailes) para que los estudiantes ejecuten las acciones descritas conforme a cómo surgen las tarjetas.
Actividades con recursos didácticos tangibles
Son actividades que involucran objetos tangible para representar un problema y avanzar su posible solución. Un ejemplo de este tipo de actividad es el juego tangram y otros tipos de rompecabezas semejantes. En el nivel inicial, y como una analogía, puede citarse la

caja de recursos didácticos de Froebel (Fröbelgaben) donde los estudiantes aprenden propiedades sobre los cuerpos y formas a partir de interactuar con piezas de madera (Manning 2005). Otro ejemplo, más cercano al desarrollo del PC son las plataformas digitales interactivas que se configuran sin la necesidad de tener asociadas una computadora, y que responden a órdenes dadas mediantes botoneras o tarjetas (McNerney 2004). Puede ser discutible el considerar a este último ejemplo como una estrategia desconectada, dado que si bien no utilizan computadoras sí involucran tecnologías digitales.
Juegos de mesa
En esta categoría se encuentran aquellos juegos de mesa que se valen de diferentes elementos para plantear problemas propios de ciencias de la computación. En estas actividades los estudiantes participan en calidad de jugadores desarrollando capacidades del PC a la medida que progresan en sus partidas de juego. Un ejemplo de este tipo de juegos lo constituye "Code & Roby"[6], que ha sido probado con éxito en escuelas de nivel primario (Ferrari, Rabbone y Ruggie, 2015). Este juego implementa una versión "desconectada" de la clásica Tortuga Logo (Ferrari y otros, 2015).
Actividades de problemas donde interviene la lógica
Son actividades que se centran en desarrollar el razonamiento lógico, en particular utilizando los operadores lógicos (conjunción, disyunción, negación, comparaciones, etc.) en situaciones problema. En general, se suelen presentar en la forma de enunciados que expresan restricciones que se deben cumplir para lograr un objetivo. Suelen incluir estructuras de decisión. Un ejemplo de actividades: "Si Ángela habla más bajo que Rosa y Celia habla más alto que Rosa, ¿habla Ángela más alto o más bajo que Celia?"
Actividades de identificación de patrones
Estas actividades buscan trabajar las habilidades relacionadas con el descubrimiento de patrones y la capacidad de crear generalizaciones y abstracciones. Suelen ser sencillas y basarse tanto en elementos gráficos como en las características de una serie de elementos a los que hay que aplicarles una etiqueta o separarlos en alguna categoría. Las actividades pueden incluir tareas para descubrir las reglas que subyacen a la división de dos o más conjuntos de elementos. Un ejemplo podría ser buscar que diferencia al conjunto "{manzana, cerezas, frutilla}" de "{limón, banana, melón}" (en este caso el color). El proyecto Bebras, que se enfoca en evaluar y desarrollar el PC en los niños y adolescentes, utiliza este tipo de actividades tanto en el desafío Bebras (una prueba internacional que permite medir el nivel del desarrollo del PC en los jóvenes) como en las tarjetas que propone para trabajar estos temas en el aula (Izu y otros, 2017).

6 http://code.intef.es/cody-roby/

Actividades de ejecución de algoritmos
En este caso son actividades donde a partir de la explicación de cómo funciona un algoritmo, los estudiantes deben ejecutarlo sobre un conjunto de datos. Ejemplos de este tipo de actividad pueden ser construir imágenes en grillas a partir de reglas o bien aplicar algoritmos de ordenamiento a un conjunto de cartas numeradas. Suelen utilizarse en todos los niveles educativos incluyendo el nivel universitario (Beecher, 2017), ya que permiten explicar el funcionamiento de algoritmos de diferente complejidad y constituyen un paso previo a la creación de programas de computadoras.
Actividades de descubrimiento de algoritmos
Son actividades que buscan que los estudiantes descubran cuál es el algoritmo que hay detrás de alguna tarea. Un ejemplo común de este tipo se constituye la práctica habitual en cursos de programación donde se explica cómo funciona la búsqueda binaria (buscar un elemento en una lista ordenada a partir de sucesivas comparaciones por mayor y menor) a partir de una secuencia de tarjetas con números que se presentan de forma ordenada. Su diferencia con la categoría anterior reside en que en este caso el algoritmo no se les presenta a los estudiantes explícitamente sino que debe surgir a partir de trabajar con la tarea que se les ha presentado.
Actividades de creación de algoritmos
Son actividades similares a las anteriores, pero que corresponden a problemas que tienen soluciones más amplias y donde la creatividad puede tener un papel más importante -donde se aplica el pensamiento divergente (Hocevar 1980)-. Son cercanas a la programación pero no incluyen el uso de computadoras, por lo que requiere que una persona opere como agente y ayude a interpretar las instrucciones para realizar la tarea especificada. Son apropiadas para realizar en grupos y funcionan en diferentes niveles educativos dependiendo de la complejidad. Es una práctica habitual en la enseñanza de la programación crear programas en “pseudocódigo” (basado en el lenguaje natural, de un alto nivel de abstracción entendible por las personas y no por una computadora) como un paso previo a la creación de un programa en computadora. Otro ejemplo de creación de algoritmos con herramientas desconectadas pueden ser las secuencias creadas con las tarjetas para que sean ejecutadas en el juego “Cody & Ruby” .

A partir de esta clasificación se espera poder contribuir al estudio y desarrollo de las actividades desconectadas. Asimismo, gracias a la investigación realizada del estado del arte de este tipo de estrategias didácticas, hemos podido construir algunas propuestas en esta misma línea, las cuales se presentan en el siguiente apartado. El objetivo de estas actividades se enfoca en desarrollar experiencias de aula que permitan trabajar, de forma integra, las diferentes dimensiones del PC.

Colección UNIPE - Actividades Desconectadas

Como parte del proyecto educativo “Saberes Digitales” de la Universidad Pedagógica Nacional (Bordignon, 2019) se presentan tres actividades, de tipo desconectado, que inauguran una colección de recursos didácticos[7] que tienen por objetivo colaborar en la conceptualización y el desarrollo de capacidades en torno al pensamiento computacional en los niveles primario y secundario.
En el proceso de diseño de las actividades, se ha tenido en cuenta la experiencia y los aprendizajes del proyecto pionero CS Unplugged, así también como los criterios de diseño de estas actividades (Tomohiro y otros, 2009) y otras propuestas similares (Tsarava y otros, 2018). Además se consideraron las recomendaciones que utiliza el equipo creador de las tarjetas Bebras (Dagiani y Futschek, 2008). Las actividades de la colección que se presentan toman como eje central de desarrollo al pensamiento algorítmico y mantienen un diálogo permanente con las otras dimensiones que integran el pensamiento computacional (Futschek y Moschitz 2010). Entendemos que esta forma de trabajo permite ofrecer materiales atractivos para los estudiantes al trabajar sobre todas las dimensiones mencionadas de forma conjunta.

El primer recurso didáctico de la colección se denomina “La hamburguesería de Don Simón” y está dirigido al segundo ciclo de la escuela primaria. El foco de la propuesta está en desarrollar el concepto de instrucción, secuencia, algoritmo y programa, y también brevemente aborda una introducción a las instrucciones de tipo condicional. Se utiliza como metáfora de trabajo una máquina hipotética que confecciona hamburguesas con diferentes ingredientes (Figura 1). Los problemas que deben resolver los estudiantes están en relación con el diseño de distintos programas para dar solución a una variedad de pedidos de clientes. Para ello deben confeccionar algoritmos y traducirlos a un lenguaje de máquina de cocina. La secuencia de problemas avanza en complejidad a medida que se realizan actualizaciones al menú ofrecido y a la máquina. Como actividad final, se propone a los estudiantes que creen su propia artefacto y las instrucciones asociadas para servir las bebidas del establecimiento. El objetivo de esta última parte, es que los estudiantes enfrenten un desafío en el cual tengan más libertad al momento de crear las posibles soluciones.

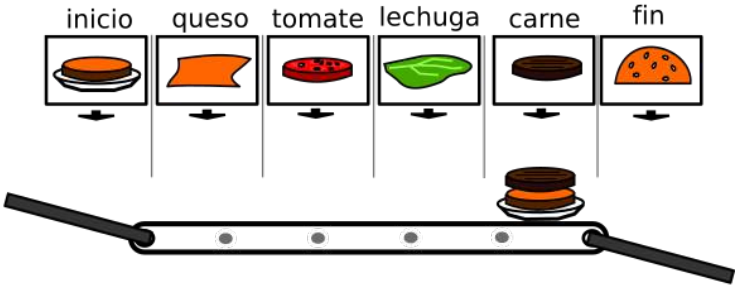


Figura 1. Máquina hipotética que funciona hamburguesas

7 http://saberesdigitales.unipe.edu.ar/

La segunda actividad propuesta consiste en un juego inspirado en la Tortuga Logo y guarda similitudes con el jueg “Cody & Roby”. La actividad se denomina “Carrera de ratones” y puede trabajarse tanto en el nivel primario como en los primeros años del secundario. El foco de esta propuesta está en el desarrollo del concepto de instrucción, secuencia, estructura repetitiva y programa. El juego permite que dos o más jugadores compitan para capturar la mayor cantidad de quesos que se presentan sobre un tablero. Para lograrlo los participantes deben crear programas, utilizando las tarjetas que se reparten de forma aleatoria y así guiar a sus ratones (ver figura 2). Cada programa creado es ejecutado por el contrincante, para así fomentar el desarrollo de la capacidad de evaluación. Gracias al azar, los jugadores se encuentran con nuevas situaciones en cada partida por lo que no solo se deben comprender los conceptos claves de programación (existen tarjetas de repetición para experimentar con bucles) sino que también deben utilizar su ingenio y creatividad para sacar el mejor partido de sus cartas.

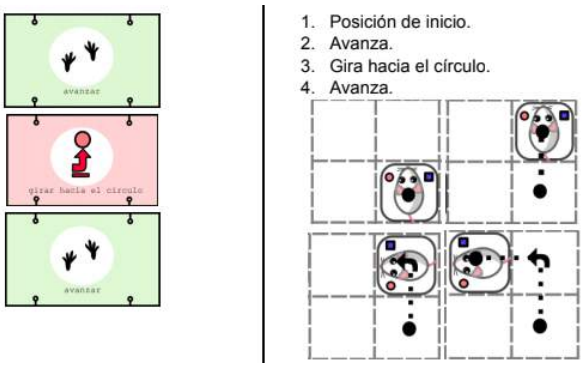


Figura 2. Ejemplo de programa y ejecución con un ratón

El último recurso de la colección, es denominado “Los desafíos del ratón” y está orientado tanto al nivel de enseñanza de primaria como a los primeros años de la secundaria. El objetivo de estas actividades es avanzar sobre el pensamiento algorítmico, trabajando con mayor profundidad las estructuras secuenciales, condicionales y repetitivas. La metáfora utilizada tiene que ver con niveles de un videojuego que van avanzando en complejidad a partir de introducir instrucciones nuevas. Estos desafíos brindan la oportunidad de desarrollar optimizaciones (ver figura 3), ya que se le solicita a los estudiantes que creen los programas que utilicen la menor cantidad de tarjetas posibles. El objetivo en éste último punto es promover las capacidades de abstracción, generalización y pensamiento en términos de evaluación.

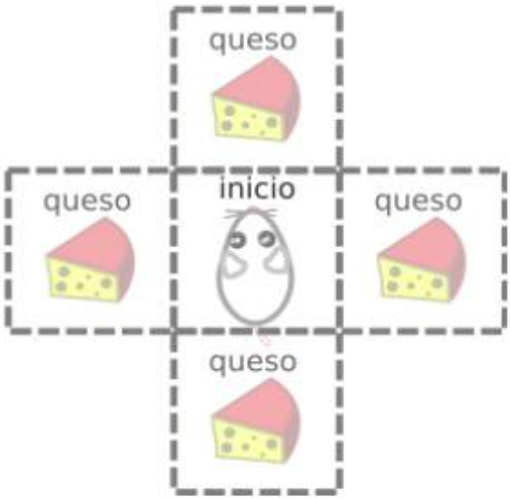


Figura 3. Ejemplo de desafío de ratón

Los tres recursos mencionados se encuentran en una etapa de pruebas y refinamiento. Se está trabajando con grupos de maestros y profesores en instancias de capacitación y de prueba de materiales en aula. Se entiende que para lograr recursos educativos maduros, que enriquezcan la clase, es necesario un proceso largo, de ida y vuelta, de pruebas y reflexiones.

Consideraciones finales

A partir de la inclusión de nuevos NAPs en la educación argentina, se abren nuevas puertas y oportunidades para trabajar la resolución de problemas a partir del apoyo dado por el pensamiento computacional. Las estrategias didácticas desconectadas se presentan como una opción válida para comenzar este camino gracias a que, en principio, no requieren de recursos tecnológicos especiales. Sin embargo, el marco conceptual que las sustenta se encuentra aún en una etapa de construcción, y un aporte de este trabajo está en relación con la presentación de una taxonomía que colabore en su entendimiento y desarrollos futuros.

Como segundo aporte, se ha presentado una colección de actividades de elaboración propia, que buscan colaborar con el desarrollo de experiencias relacionadas al PC, en particular vinculadas al pensamiento algorítmico. A partir de lo trabajado se configuran nuevas líneas de investigación, en base a que reconocemos que es necesario avanzar en el diseño de prácticas más complejas que ayuden a resolver problemas asistiéndose en el pensamiento divergente.

Como trabajo futuro se plantea continuar con el refinamiento de la taxonomía propuesta así también como con la elaboración de propuestas didácticas que permitan trabajar el PC de forma desconectada. Finalmente, y como parte de otro horizonte de trabajo, se pretende avanzar en propuestas que permitan realizar una transición didáctica entre este tipo de actividades y aquellas que incluyen la tarea de programación con herramientas digitales.

Bibliografía

Beecher K.(2017) Computational Thinking. *BCS Learning & Development* LTd 2017.

Bell, T. Alexander, J. Freeman, I, Grimley, M. (2018) Computer science unplugged: school students doing real computing without computers. *International Journal of Serious Games*. Volume 5, Issue 2. June 2018

Bell T - Vahrenhold J (2018). CS Unplugged—How Is It Used, and Does It Work? *Springer Nature* Switzerland AG

Bell T. Witten I. Fellows. M (2008) Computer Science Unplugged: Un programa de extensión para niños de escuela primaria. *Computer Science Unplugged.*

Bordignon, F. (2019) Saberes Digitales en la Educación Primaria y Secundaria de la República Argentina. *Espiral, Revista de Docencia e Investigación.* (en prensa)

Consejo Federal de Educación N°343/18. Disponible en: http://www.bnm.me.gov.ar/giga1/normas/RCFE_343-18.pdf

CAS (2015) Pensamiento Computacional. Guía para profesores. Computing At School.

Dagiene V. Futschek G. (2008) Bebras International Contest on Informatics and Computer Literacy: Criteria for Good Tasks. R.T. Mittermeir and M.M. Sysło (Eds.). ISSEP 2008, LNCS 5090, pp. 19–30, 2008. *Springer-Verlag* Berlin Heidelberg 2008

Ferrari F. Rabbone A Ruggiero S. (2015) Experiences of the T4T group in primary schools. International Conference on Informatics in Schools. ISSEP 2015. University of Ljubljana, Faculty of Computer and Information Science.

Futschek G. Moschitz J. (2010) Developing Algorithmic Thinking by Inventing and Playing Algorithms.Constructionism 2010, Paris

Hocevar D(1980), Intelligence, divergent thinking, and creativity. Intelligence, Volume 4, Issue 1, Pages 1-95 (January–March 1980)

Izu C. Mirolo C. Settle A. Mannila L. Stupuriene G. (2017) Exploring Bebras Tasks Content and Performance: A Multinational Study. Informatics in Education, 2017, Vol. 16, No. 1, 39–59 © 2017 Vilnius University.

Kotsopoulos Donna & Otros (2017) A Pedagogical Framework for Computational Thinking Digit Exp Math Educ. *Springer International Publishing*.

Kumar, D. (2014). Digital playgrounds for early computing education. ACM Inroads, 20–21.

Manning J. (2005) Rediscovering Froebel: A Call to Re-examine his Life & Gifts. *Early Childhood Education Journal*, Vol. 32, No. 6, June 2005

McNerney, T. (2004) From turtles to Tangible Programming Bricks: explorations in physical language design. Pers Ubiquit Comput t (2004) 8: 326–337.

Ozcinar, H., Wong, G., & Ozturk, H. T. (Eds.). (2017). Teaching Computational Thinking in Primary Education. IGI Global.

Taub, R., Armoni, M., y Ben-Ari, M. 2012. CS unplugged and middle-school students' views, attitudes, and intentions regarding CS. *ACM Trans. Comput. Educ.* 12, 2, Article 8 (April 2012), 29 pages.

Tissenbaum M. Sheldon J, Abelson Hal. From Computational Thinking to Computational Action (2019). Communications of the ACM. Vol 62. No 3. Marzo 2019.

Tomohiro N. Kanemune S. Idosaka Y. Namiki M. Bell. T. Kuno. Y (2009). A CS Unplugged Design Pattern. SIGCSE'09. Chattanooga, Tennssee, USA. ACM.

Tsarava K. Moeller, K. Ninaus M. (2018) Training Computational Thinking through board games: The case of Crabs & Turtles International Journal of Serious Games. Volume 5, Issue 2. Junio 2018.

Wing, J. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

Potencialidad didáctica de las actividades experimentales abiertas para la enseñanza de disciplinas STEM

Víctor Furci, Oscar Trinidad, Fernando Bordignon, Luis Peretti[1]

Resumen: En este trabajo se presenta, desde una perspectiva cualitativa de estudios de caso, una indagación sobre la potencialidad didáctica y formativa de secuencias didácticas para la enseñanza de disciplinas STEM (Science, Technology, Engineering and Mathematics), centradas en el diseño e implementación de actividades experimentales abiertas para enseñanza de las ciencias, que integren tecnologías digitales Arduino y su programación. Las secuencias, diseñadas por el grupo de investigación, se ofrecen a un conjunto de docentes en ejercicio, tanto en el nivel secundario, como en instituciones de formación docente. Se propone a estos docentes el análisis previo, adaptación, implementación y evaluación de las secuencias elaboradas, con la finalidad de producir elementos de análisis didáctico sobre las articulaciones de las disciplinas involucradas (Física, Biología, Electrónica, Programación, Matemática, entre otras). Las secuencias ofrecidas se elaboran teniendo en cuenta los resultados de estudios previos desarrollados por el grupo de investigación (Furci, Trinidad y Peretti, 2018), y se caracterizan por proponer el abordaje sistémico de situaciones problemáticas significativas desde una perspectiva CTSA (Pérez, 2013), promover el diseño e implementación de actividades experimentales, la integración de dispositivos basados en la tecnología Arduino y su programación, y presentar un amplio grado de apertura desde la perspectiva STEAM (Bybee, 2010). El trabajo surge, además, como continuidad del proyecto de innovación "Más Allá de las Pantallas", centrado en propuestas de diseño e implementación de objetos interactivos digitales como medio para el desarrollo de la fluidez digital (Bordignon e Iglesias, 2015). Se presentan resultados, obtenidos utilizando metodología de clínica didáctica (Rickenmann, 2007) y técnicas de análisis del discurso (De Longhi, 2012), relacionados a tres dimensiones principales: la potencialidad didáctica de las secuencias didácticas analizadas, los saberes profesionales puestos en juego por los docentes, y algunas reflexiones sobre el tipo de aprendizaje desarrollado por los estudiantes. También se incluyen algunos aportes destinados al desarrollo de propuestas de formación docente inicial y continua que articulen saberes propios de las distintas disciplinas del área.

Palabras clave: Enfoque STEM – Interdisciplina - Didáctica de las ciencias naturales.

1 Universidad Pedagógica Nacional.

Problema de investigación y marco teórico

Algunas de las dificultades para la enseñanza de elementos básicos de programación que son habitualmente informadas en trabajos de investigación en el área (Szpiniak, 2006; Fundación Sadosky, 2013) consisten en su presentación desarticulada, compartimentalizada y descontextualizada a los estudiantes, cierta tendencia excesiva hacia la conceptualización y generalización algorítmica por sobre la práctica concreta de resolución de problemas y la falta de articulación con otras disciplinas abordadas en los distintos diseños curriculares.

En este sentido, consideramos que el trabajo que nuestro grupo de investigación viene desarrollando durante los últimos 6 años, relacionado a la formación inicial y continua de docentes de ciencias experimentales, puede realizar aportes de interés. Durante el bienio 2016-2017, nuestra investigación se centró en el proceso de integración curricular de tecnologías digitales (particularmente el desarrollo de la computación física con placas Arduino y su entorno de desarrollo) en las actividades experimentales de las aulas de formación docente inicial de los profesorados de Ciencias Naturales. Aunque en forma estricta, esta temática no es considerada parte de la didáctica de la programación, constituye un escenario potencialmente didáctico para su análisis y desarrollo. Por otra parte, reviste actualidad e interés en el marco de las transformaciones educativas relacionadas a la enseñanza de la programación en todos los niveles educativos, expresadas en las resoluciones del Consejo Federal de Educación vinculadas a los Núcleos de Aprendizaje Prioritarios (CFE, 2018) y al Marco de Organización de los Aprendizajes (CFE, 2017).

Por otra lado, la temática se inscribe en el desarrollo de tendencias internacionales, vinculadas a la enseñanza de las ciencias experimentales en general y de la Física en particular, que se vienen implementando en numerosos países y, que en muchos de los casos, toman como cuestión nodal la integración curricular de las diversas disciplinas científicas junto con la tecnología, la ingeniería y la matemática en el modelo didáctico iniciado en Estados Unidos en la década del 90´, y luego extendido a otros países, bajo la denominación general de STEM (*Science, Technology, Engineering and Mathematics*).

Si bien algunos autores enfatizan sobre la potencialidad didáctica del modelo STEM (Araya, 2016; García, 2017), otros alertan sobre el poco desarrollo y logros de estas propuestas (Bybee, 2010; Brown, 2012), y sobre cierta tendencia, algo ingenua, relacionada con el “solucionismo tecnológico” atribuido al pensamiento computacional (Adell, 2019).

En el contexto de estos debates resulta de gran importancia la reflexión sobre la potencialidad didáctica del modelo STEM en este momento histórico, en el que numerosos programas de equipamiento tecnológico a nivel internacional suelen intentar ingresar al sistema educativo de forma indirecta, sin considerar la opinión de docentes, especialistas en didáctica o miembros de la comunidad educativa. Como ejemplo de estos procesos podemos analizar el creciente impacto de los programas STEM en América Latina (Peretti y otros, 2019). Si bien algunos de estos programas presentan ciertos puntos de interés, en

relación a la promoción de la innovación, la creatividad, la importancia del diseño, la apertura a diversas tecnologías (no limitadas a un determinado kit de robótica, como modelos predefinidos para armar), presentan numerosas limitaciones en relación a su fundamentación epistemológica, pedagógica y didáctica. En este sentido resultan de gran interés los aportes de una línea recientemente presentada como "*Nature of Science, TechnologyEngineering, Art and Mathematics*" NOSTEAM (Ortiz-Revilla, Adúriz-Bravo y Greca, 2020), que propone incluir en el desarrollo de proyectos STEM, aportes y reflexiones epistemológicas en relación a la naturaleza de la ciencia. En cuanto a la noción de arte, se apunta al sentido amplio de las "artes liberales" de la antigüedad clásica, para revitalizar una formación humanista e integral de los profesores de ciencias.

Se trata de abordar no solamente los aspectos formales y técnicos de estas disciplinas científicas, sino también su "naturaleza", en el sentido de los contextos y procesos de producción y validación de los saberes producidos en cada una de ellas.

Otro aspecto central relacionado con la viabilidad de implementación de este tipo de propuestas consiste en la selección y diseño de problemas adecuados, coherentes con esta línea didáctica. Se trata de diseñar y plantear buenos problemas que resulten adecuados para los estudiantes, en términos de interés y motivación, relevancia social, alto grado de contextualización, nivel de complejidad adecuado, riqueza conceptual disciplinar, potencialidad para promover reflexión sobre la naturaleza de la ciencia y sus contextos de producción, grado de apertura, originalidad, posibilidad concreta de ser resueltos por los estudiantes en forma genuina y honesta, con los tiempos y recursos disponibles, entre otras características.

Particularmente en investigaciones previas desarrolladas por nuestro grupo (Gutiérrez y otros, 2017; Furci y otros, 2018, Trinidad y otros, 2019), observamos que las dificultades manifestadas por los docentes en relación a la integración curricular de la tecnología en la enseñanza de las ciencias experimentales son de diversa índole: especificidad y costo de los equipamientos necesarios, escasa tradición didáctica en la implementación de proyectos de investigación escolar, problemas técnicos relacionados con el diseño y programación de dispositivos, falta de conocimientos didácticos específicos para la enseñanza de las disciplinas involucradas, etc. Resumiendo, podemos decir que la implementación concreta de proyectos STEM en aulas de Física, se encuentra dificultada por la debilidad de los conocimientos tecnológicos, curriculares y didácticos del contenido a enseñar (Park, 2007) que los docentes deben construir, generalmente en forma solitaria, en tiempos escolares que lo condicionan.

Por otra parte existen aspectos simbólicos subyacentes sobre lo tecnológico en docentes y estudiantes que de alguna manera condicionan las acciones concretas de implementación didáctica de tecnologías. En particular, las representaciones sociales (Moscovici, 1969 -citado en Piña Osorio y Cajiga 2004; Di Giacomo, 1987) de los actores involucrados en los procesos de enseñanza y de aprendizaje constituyen un sistemas simbólico con fuerte incidencia en la implementación de estrategias de integración tecnológica.

Otra cuestión no menor, son las escasas publicaciones de materiales y propuestas de aula, que incluyan un análisis didáctico y que, desde esa perspectiva, pudieran servir como orientación para la formación docente. Entendemos que para un avance en las posibilidades de implementación real de proyectos STEM en las aulas, no solo hace falta avanzar en el desarrollo de propuestas integradoras que den respuesta a los problemas señalados, sino que estas propuestas deben ser analizadas desde su implementación en contextos reales de desempeño, caracterizando así su potencialidad didáctica para la enseñanza de la disciplinas involucradas, analizando en qué medida promueven el aprendizaje de los contenidos seleccionados y el desarrollo de habilidades cognitivas de orden superior en los estudiantes.

Todas estas dificultades, conforman una *brecha* entre las prácticas deseables y pensadas desde marcos teóricos como el STEM y las posibles para un gran número de docentes que tienen que salvar estos obstáculos generalmente por cuenta propia.

Una forma de acortar esta brecha, es establecer nexos o puentes entre aspectos centrales de las didácticas específicas más consolidadas (como la didáctica de las ciencias naturales y las matemáticas) con las propuestas didácticas propias de las ciencias de la computación, que se encuentran en una etapa inicial de su desarrollo. Nuestra propuesta selecciona y prioriza algunas líneas de investigación con tradición en la didáctica de las ciencias naturales, como son la enseñanza por medio de la modelización (elaboración de modelos), contextualización (enseñanza basada en contextos) y reflexiones sobre la naturaleza de las ciencias, vinculada a la integración de tecnología en la enseñanza de las ciencias.

En relación a la didáctica de la programación, se pretende centrar el análisis sobre el enfoque o paradigma de programación seleccionado por los docentes (modelos procedimentales, orientado a objetos o funcionales) los contenidos abordados (elementos de programación básica como por ejemplo: tipos de datos, definición de variables, secuencias lineales, condicionales, ciclos, funciones y almacenamiento de datos) y el alcance en este tipo de propuestas (considerada inicialmente como una alfabetización básica en programación, vinculada a la configuración y adaptación de programas desarrollados por otros autores) (Fundación Sadosky, 2013).

La enseñanza basada en la modelización (Galagovsky y Adúriz-Bravo, 2001), propone adoptar un concepto de modelo científico que podemos denominar semántico (Adúriz-Bravo, 2010), en el que las diversas formas de representación (imagen analógica, análogo concreto, ecuaciones, texto, simulaciones digitales, etc.) sean igualmente válidas para la construcción de conocimiento científico escolar, en la medida que permitan describir, explicar, predecir e intervenir sobre el mundo natural (en sistemas o procesos).

Aunque “contexto” es un término polisémico (Zapata, 2016), a grandes rasgos podríamos resumir su importancia en el campo de la enseñanza de las ciencias considerando la Enseñanza Basada en Contexto (EBC) como una metodología consistente en construir y desarrollar conocimientos científicos a partir de situaciones “similares” a las del mundo real.

Estas situaciones se usan como estructura central para ir introduciendo los conceptos científicos a medida que son necesarios y desarrollar así una mejor comprensión de los problemas planteados (King y Richtie, 2012). Inicialmente, la EBC toma un contexto como punto de partida para introducir un tema y activar la curiosidad de los estudiantes, logrando mejorar el interés, la motivación y la actitud hacia la ciencia, promoviendo una mayor participación de los estudiantes al incrementar su satisfacción personal y aumentar la motivación. Pero además el trabajo en contexto, debiera concebirse como una práctica que otorga sentido al contenido, porque lo vuelca a la resolución de problemas específicos relacionados con la ciencia y la tecnología, y lo pone en situación de "modelo", aplicando un procedimiento para la resolución del problema, que comporta aprender el conocimiento científico tecnológico, habilidades y actitudes necesarias para resolverlo (Kortland, 2007). Podríamos decir que el objetivo central de este enfoque es potenciar un aprendizaje más significativo de las ideas científicas, a partir de facilitar las conexiones teoría-realidad, y atenuar la separación entre ambas, propia de la enseñanza tradicional (Moraga Toledo y otros, 2019).

Desde una perspectiva pedagógica y didáctica adscribimos a un contexto de inclusión tecnológica que denominamos "Modelo de Integración Tecnológica" (MIT). Este modo de entender la inclusión tecnológica en educación a través de los diversos programas o iniciativas de alfabetización tecnológica y digital, supone una perspectiva emancipadora de la ciudadanía. En sus propósitos, se promueven una perspectiva de empoderamiento de las tecnologías para fortalecer la utilización crítica de estas, construir un enfoque ético acerca de su utilización, promover el diseño y la creatividad en el desarrollo de soluciones tecnológicas con compromiso social y ambiental. Apuntan al desarrollo, utilización y diseño de software de código abierto, comunidades colaborativas y aplicaciones que promuevan el abordaje de problemas sociocientíficos y la participación ciudadana.

Las dimensiones imbricadas en esta concepción -más allá de su perspectiva política que como finalidad persigue la distribución social del conocimiento para fortalecer el tejido comunitario, la organización democrática y la participación individual y colectiva soberana en la toma de decisiones-, también se apoyan en una manera de entender la ciencia, el quehacer científico y las claves para su enseñanza.

Desde una perspectiva didáctica, creemos que una propuesta MIT, debe desarrollarse en torno al trabajo con problemas abiertos y complejos, del ámbito sociocientífico, que requieren el diseño de dispositivos experimentales (Furci y otros, 2019; Trinidad y otros, 2019) y el diseño y construcción de objetos interactivos digitales (Bordignon e Iglesias, 2015). En este tipo de propuestas se promueve la construcción de conocimientos contextualizados, articulados en modelos explicativos significativos para los estudiantes y validados con criterios que se ajustan a una perspectiva epistemológica actualizada (Adúriz-Bravo, 2018), que permitan describir, explicar, predecir e intervenir sobre los fenómenos naturales en estudio.

Finalmente, y con respecto a las propuestas didácticas, nos interesa estudiar el concepto de potencialidad didáctica, el cual es tomado del modelo didáctico de investigación escolar, presentado por el grupo IRES (Porlán Ariza y García Pérez, 2000) en donde la clase es considerada como un sistema, en el que los flujos de información son permanentes, como consecuencia de la diversidad de interacciones simultáneas, de distinta intensidad que se dan entre alumnos, profesores, materiales didácticos, contexto físico, etc. y aportan al sistema aula una determinada organización, de la que emergen unas cualidades y potencialidades propias de cada situación en contexto. Así descrita, el aula constituye, sobre todo, un sistema de comunicación. Por último, una secuencia será potencialmente didáctica si de alguna manera presenta posibilidades para generar estos diversos tipos de interacciones dentro de la clase de Física.

Otros elementos que aportan al concepto de potencialidad didáctica provienen de didáctica específica (Zoller y Scholz, 2004) y se centran en relevar los tipos de habilidades cognitivas puestas en juego en la resolución de distintos tipos de actividades. Así entonces se diferencian los pensamientos de orden superior (HOCS) en relación con las más tradicionales de orden inferior (LOCS). Teniendo en cuenta esta perspectiva, las propuestas de formación docente deben considerar y poner en juego, de forma equilibrada, los conceptos físicos específicos, sus aplicaciones y relaciones, los procesos de medición, registro y cálculo como así también las cuestiones relacionadas con los impactos sociales, ambientales y económicos de las cuestiones tratadas.

Desde esta posición pedagógica y didáctica, algunas de las preguntas que intentamos responder en el presente trabajo, y caracterizan el problema de investigación son las siguientes:

1. ¿Qué características deben tener los materiales didácticos que andamien efectivamente la implementación concreta de proyectos STEM en aulas de las disciplinas involucradas y promuevan habilidades cognitivas de orden superior?
2. ¿Cuáles son las valoraciones y adaptaciones que los docentes participantes pueden hacer sobre propuestas de aula diseñadas por el equipo de investigación en función de las características de los grupos y establecimientos en donde trabajan?
3. ¿Cuáles son los conocimientos que los docentes estudiados construyen en la interacción grupo investigador - muestra a lo largo del recorrido de formación propuesto?
4. ¿Qué grado de aplicación o validez tiene el modelo STEM en nuestro contexto de formación docente?
5. ¿Cuál es la potencialidad didáctica de las secuencias propuestas a la muestra?, esto es, ¿en qué sentido promueven la interacción de distinto tipo de conocimientos, actores, y disciplinas? ¿En qué medida promueven el desarrollo de habilidades cognitivas de orden superior? ¿Cuál sería el papel de la programación en proyectos de enseñanza con formato STEM? ¿Cuál es el aporte que realiza esta experiencia a la formación en ciencias de la computación?

6. ¿Hasta qué punto las características de la secuencia propuesta por el equipo de investigación (objetivos, contenidos, actividades, etc.) permiten la discusión y el aporte de los docentes, desde sus disciplinas específicas?

Objetivos de la investigación

En base a lo desarrollado en el apartado anterior, definimos el siguiente objetivo general:

1. Diseñar y poner a prueba un dispositivo de formación docente en Ciencias Experimentales (focalizando en Física en particular), centrado en el proceso de elaboración, implementación en el aula y evaluación de secuencias didácticas que incluyan actividades experimentales abiertas con tecnología ARDUINO, desde un enfoque STEM, en instituciones de formación docente.

En relación al objetivo general, planteamos los siguientes objetivos específicos:

1. Analizar y caracterizar la potencialidad didáctica de propuestas de enseñanza de Ciencias Experimentales y programación, que incluyen actividades experimentales abiertas con tecnología Arduino, desde un enfoque STEM.
2. Caracterizar y analizar el dispositivo de formación propuesto en el presente trabajo.
3. Rescatar, caracterizar y sistematizar los saberes profesionales puestos en juego por los docentes en las distintas instancias del proceso de formación propuestos.

Metodología

Se adopta un diseño cualitativo de estudio longitudinal de tres casos, constituidos por tres profesores de Física que se desempeñan en Institutos de Formación Docente que se desempeñan en la Provincia de Buenos Aires.

Se propone a estos docentes participar de un proceso que se inicia con la presentación y el análisis crítico de una secuencia de enseñanza de Física que incluye actividades experimentales abiertas, que integren tecnología Arduino (diseño y programación de dispositivos) desde una perspectiva STEM. El proceso continúa con la adaptación, por parte de los docentes de la muestra de dicha secuencia a su contexto de desempeño, su implementación, finalizando con el análisis de su práctica y evaluación de los aprendizajes.

La secuencia compartida con los docentes inicia con un problema contextualizado en situaciones de la vida cotidiana, en particular, los problemas para transitar el invierno que sufren los habitantes de asentamientos de bajos recursos que construyen sus viviendas con materiales recolectados en la vía pública. El problema en que se basa la secuencia consiste en proponer a los estudiantes el diseno y puesta a prueba de pequenos prototipos de viviendas realizadas con materiales de descarte, con el fin de optimizar su rendimiento termico. La evaluacion experimental del rendimiento termico de los prototipos se realiza a partir de la obtención de un gráfico de temperaturas en función de tiempo, al incorporar una

fuente de calor en los prototipos. Se propone la utilización de tecnología Arduino para medicion, registro y análisis de las temperaturas, en distintos puntos de los prototipos, en función del tiempo.

Las dimensiones de análisis utilizadas a lo largo del proceso (presentación - crítica de la secuencia - reformulación - implementación - análisis) tienden a caracterizar tres objetos principales, fuertemente relacionados entre si:

1. La secuencia didáctica propuesta por los docentes (luego de reformular la presentada por el grupo de investigación) en relación a su potencialidad didáctica, esto es, su grado de apertura, el grado de integración curricular de tecnologías digitales, su caracterización desde la perspectiva STEM y desde las pautas de análisis de la didáctica específica de la Física y de la programación. En relación a la didáctica de la programación, se pretende centrar el análisis sobre el enfoque o paradigma de programación seleccionado por los docentes (modelos procedimentales, orientado a objetos o funcionales) los contenidos abordados (elementos de programación básica como por ejemplo: tipos de datos, definición de variables, secuencias lineales, condicionales, ciclos, funciones y almacenamiento de datos) y el alcance en este tipo de propuestas (considerada inicialmente como una alfabetización básica en programación, vinculada a la configuración y adaptación de programas desarrollados por otros autores) (Fundación Sadosky, 2013).

2. El valor formativo para los docentes en relación a los aprendizajes y saberes profesionales puestos en juego y desarrollados a lo largo del proceso analizado, sus reflexiones, conceptualizaciones y acciones didácticas consideradas desde el modelo de “Conocimiento Didáctico del Contenido” (Park, 2007). En particular, a las concepciones docentes sobre las Ciencias de la Computación y su didáctica, el modo en que los docentes gestionan las articulaciones de disciplinas concurrentes, revalorizando el rol del profesor de programación en el trabajo colaborativo para el desarrollo de este tipo de proyectos.

3. Los aprendizajes de los estudiantes, vinculados a los contenidos conceptuales de Física y poniendo en valor los aportes de las ciencias de la computación, a la resolución de este tipo de problemas. También se aborda el análisis desde la perspectiva del desarrollo de habilidades cognitivas de orden superior (HOCS) vinculados a los ambientes de aprendizaje generados en la propuesta didáctica (Zoller, 2013).

Para el análisis de las secuencias didácticas reformuladas por los docentes definimos la dimensión, potencialidad didáctica de la secuencia, ya trabajada en el marco teórico. Entendemos que las propuestas didácticas que los docentes presentan a los estudiantes, pueden poner en marcha (o no) diversas interacciones entre todos los participantes y elementos que componen el sistema clase (alumnos, profesor, materiales didácticos,

espacio físico, materiales, etc.). En cada una de estas interacciones, existe información que se aporta, se obtiene, se reconfigura, se utiliza, etc. Proponemos que un análisis de la cantidad y calidad de estas interacciones nos brinda información sobre la potencialidad de la secuencia, esto es, si el material posibilita que estas interacciones ocurran. Así entonces, sería necesario poder anticipar posibles interacciones para analizar si las secuencias propuestas las fomentan. Con este fin, se construye un instrumento (ver Anexo 1) adaptado de Raths (1971) con las siguientes variables:

V1: Expliciten su representación sobre la tarea: Sería deseable, desde aspectos metacognitivos, que en la secuencia de aprendizaje el docente contemple algún momento o espacio en donde acordar (en el grupo clase) los objetivos y logros esperados al plantear una actividad, esto es, lograr una representación personal y grupal de la tarea consensuada. Esta variable también refiere a las formas y acuerdos que respaldan los procesos de evaluación.

V2: Desempeñen un rol activo. El tipo de actividades presentes en las secuencias didácticas pueden poseer diversos grados de apertura, en forma resumida, demandar al alumno desde el simple seguimiento de un protocolo para su resolución, hasta requerir que éste tome decisiones que condicionan la validez de la respuesta presentada por el estudiante. Estos últimos tipos de actividades (con mayor grado de apertura) suponen un alumno con un rol activo en la construcción de sus aprendizajes, mientras que en la otra punta del espectro, se encuentran alumnos capaces de reproducir la palabra del profesor o del seguimiento de pasos establecidos en la resolución de una actividad. En estos últimos casos, la tarea demanda un rol del alumno que podemos denominar pasivo.

V3: Interaccionen con objetos. En la enseñanza de las ciencias las actividades experimentales toman un papel especial, dado que se constituyen en la posibilidad concreta de estudiar ciertos fenómenos desde la interacción más o menos directa. Ahora las actividades experimentales nuevamente pueden demandar un bajo grado de participación del alumno (la comprobación de leyes, observar un fenómeno) o estar más cerca del formato de investigación escolar, donde existen múltiples cuestiones a resolver: la determinación de variables, diseño de experimentos y dispositivos, validación de supuestos modelos, etc. Este tipo de actividades pueden potenciarse a partir de la inclusión de tecnología que posibilite mejorar la obtención de datos experimentales, la construcción de modelos o el análisis de los mismos.

V4: Utilicen diversas habilidades de pensamientos. En la resolución de tareas los alumnos deben poner en juego diversas habilidades de pensamiento, desde aquellas denominadas LOCS de bajo orden cognitivo (recordar, identificar, nombrar, etc.) a procesos intelectuales de mayor orden, HOCS (argumentar, decidir, diseñar) o aplicar los conocimientos construidos en nuevas situaciones, contextos o materias.

V5: Construyan conocimiento colectivamente y lo comuniquen. La construcción conjunta de conocimiento en ciencias implica el trabajo grupal, la coordinación de esfuerzos, el debate en el diseño de experiencias, dominando el miedo al fracaso o crítica y valorando la originalidad. En todos los casos, la construcción conjunta supone encontrar una forma efectiva de comunicación, tanto hacia el interior del grupo de trabajo como hacia la comunidad, por lo tanto la producción y utilización de textos, tanto escritos como orales, que demanden diseño, debate, lectura, escritura y reescritura para asegurar su pertinencia.

V6: Trabajen en temáticas significativas con enfoques pertinentes. Los criterios de selección de conceptos a ser trabajados en las secuencias pueden ser diversos. Proponemos que aquellas que involucren no solo conceptos centrales de la Física, sino que expresan relaciones de los mismos con otras áreas del conocimiento, como problemáticas socioambientales, enfoque CTSA, de contexto filosófico o histórico, relaciones con aplicaciones tecnológicas, u otros elementos cercanos a las realidades de los estudiantes, permitirán más sencillamente a los alumnos establecer relaciones transformándolos en aprendizajes significativos. En particular en relación a las ciencias de la computación nos interesa analizar el enfoque didáctico seleccionado, los contenidos abordados, y el alcance de la propuesta.

V7: Desarrollen sus propuestas en tiempos, espacios y con los recursos disponibles. El conjunto de actividades previstas se desarrolla en el contexto escolar real, en tiempos y espacios disponibles, con los recursos accesibles. También se hace referencia al clima o ambiente del aula que potencia u obstaculiza la construcción de conocimiento.

Las variables y categorías de la dimensión Potencialidad didáctica, constituyen el instrumento o base de orientación, para poder valorar desde una mirada didáctica, las secuencias didácticas. No solo desde la enunciación de lo que esperamos como profesores, sino desde las observaciones de “lo que pasa” en el aula cuando estas propuestas se llevan a cabo.

No se constituye en un instrumento para “decir lo que está mal”, sino para que cada docente pueda tener una mirada de lo que pasa en su clase, y si él lo requiere, repensar reformulaciones de sus mismas secuencias teniendo en cuenta los distintos componentes que se describen en el instrumento. Detalles de la aplicación de este instrumento pueden consultarse en el trabajo presentado en el congreso Lassera 2019 (Trinidad y otros, 2019).

Con relación al Valor formativo para los docentes, la recolección, organización y análisis de los datos respeta una lógica ensayada por los autores en trabajos anteriores (Gutierrez, 2018), en los que se procede al análisis de las prácticas docentes en general y a las prácticas de enseñanza en particular, adoptando para ello elementos de la metodología de la clínica didáctica (Rickenmann, 2007). Esta metodología propone cuatro instancias principales:

1. Una entrevista previa realizada a los docentes en las instancias de presentación, crítica y reformulación de la secuencia original, anteriores a la implementación de sus clases y registro de los intercambios establecidos entre los docentes y el equipo de investigación.
2. La observación y registro de clases (realizadas a partir de video filmaciones de las clases dictadas por los docentes).
3. La entrevista de autoconfrontación (entrevistas finales, donde cada docente reflexiona sobre episodios seleccionados por el equipo de investigación de las filmaciones de sus clases), adaptado del modelo de análisis de las prácticas de Rickenmann (2007) y Nunziati (1990). Es importante aclarar que también se considerarán, en esta instancia, el registro de conversaciones e intercambios con los docentes que pongan de manifiesto su nivel de conceptualización y reflexividad, referidos a las producciones de los alumnos, el grado de apertura y potencialidad didáctica de la propuesta, los aprendizajes generados, entre otros, que van más allá del análisis de la videograbación.
4. El análisis de las producciones de los estudiantes y los resultados de los instrumentos de evaluación implementados en la secuencia.

En forma complementaria a la metodología de la clínica didáctica, y tomando como base los registros de audio obtenidos en el trabajo de campo, en uno de los docentes de la muestra, se implementaron técnicas del análisis del discurso en las clases de ciencias (De Longhi, 2012), para analizar, desde una perspectiva cualitativa, la dimensión del valor formativo para los docentes de la muestra. Estas técnicas permiten desarrollar un análisis estructural-textural (cohesión gramatical y lexical, estudio de los circuitos comunicativos de la clase, enfoques comunicativos sostenidos por docentes y estudiantes), semántico (identificando y caracterizando el patrón temático, la función de las preguntas y afirmaciones de docentes y estudiantes) y retórico (en términos de las formas de validación o autoridad puestas en juego en las interacciones discursivas). El patrón temático del discurso docente se formaliza en un esquema que se elabora analizando el tipo de conectores que utiliza el docente durante su clase y su función semántica, como por ejemplo relaciones nominales, taxonómicas, de transitividad, circunstanciales y lógicas. También se analizan las estrategias de diálogo y de monólogo que establece el profesor en el desarrollo de su clase.

Se realizó además una descripción contextual y del lenguaje no-verbal desarrollado en el caso. También se realizó un análisis de textos escritos, tanto por el docente como por los estudiantes durante el desarrollo de la secuencia. Detalles metodológicos del análisis desarrollado y resultados pormenorizados pueden consultarse en el trabajo publicado en el congreso LASERA 2019 (Furci y otros, 2019).

Finalmente en relación a los Aprendizajes de los estudiantes, se implementó un análisis detallado en el caso de uno de los cursos de la muestra. Se realizó un registro de audio de las interacciones discursivas durante la implementación de la secuencia didáctica que fue analizado por medio de las técnicas de análisis del discurso referidas en el apartado anterior (De Longhi, 2012). También se realizó el análisis de las producciones realizadas por los

grupos de estudiantes del curso a lo largo de la secuencia, adoptando técnicas de análisis del contenido (Bardín, 1991).

Resultados

Se presentan a continuación algunos de los principales resultados obtenidos, organizados de acuerdo a las dimensiones de análisis presentadas anteriormente: la secuencia didáctica, el valor formativo para los docentes y los aprendizajes de los estudiantes.

-Sobre las secuencias didácticas:

En la Figura 1 se muestran las apreciaciones correspondientes a las distintas variables que caracterizan la potencialidad didáctica de la secuencia desarrollada por el profesor A

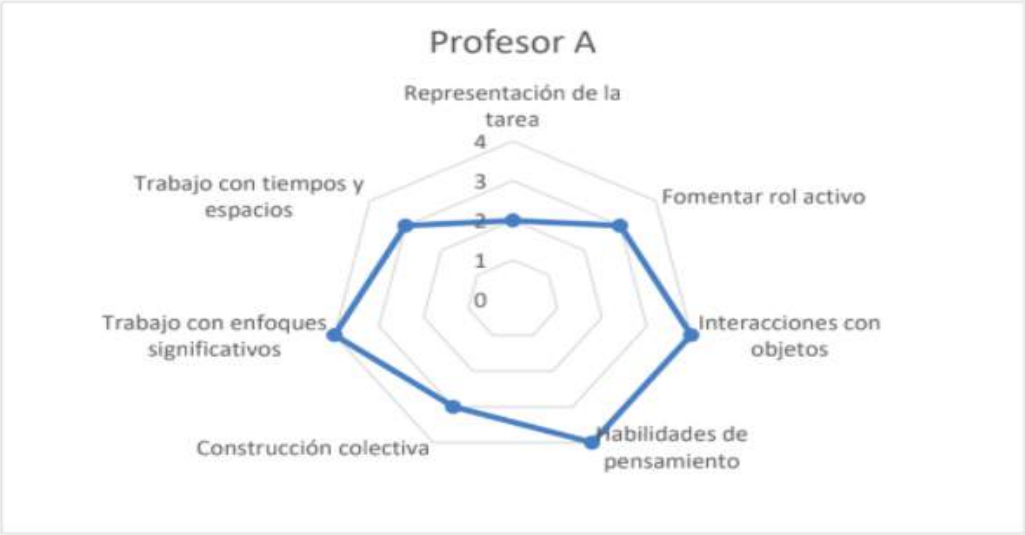


Figura 1. Fuente: elaboración propia

Los registros de la implementación de la secuencia de este profesor, nos muestran que en principio hay fuerte impacto de la componente que refiere al trabajo de los alumnos con objetos (en este caso el diseño, construcción y medición de propiedades térmicas de prototipos de casas) en conjunto con la utilización de tecnología en las mediciones. También se observa que las actividades propuestas requieren, en general, habilidades de pensamiento de orden superior. Otro aspecto importante a referir en este caso, es observar que con el importante trabajo realizado por el grupo (tres clases) el componente referido a la representación de la tarea (en relación con su evaluación) es relativamente bajo, en este caso, el docente refiere a que la formalización de lo trabajado se realizará en una prueba escrita futura, en donde se integrarán las cuestiones trabajadas con otras que aún no fueron evaluadas.

Las observaciones tomadas sobre la implementación de la secuencia didáctica del profesor B son representadas en la figura 2.

En el caso de este docente, la mayoría de las categorías poseen un desarrollo importante, posiblemente dado su alto grado de formación y experiencia. Es importante destacar que el docente se especializa en didáctica de las ciencias experimentales y tiene larga trayectoria en la formación docente continua e inicial.

Por otra parte presentaron alta valoración las actividades relacionadas con la investigación escolar, las habilidades (LOCS), los enfoques didácticos y la utilización de tiempos y espacios, que han sido trabajados dejando rastro en las actividades de sus alumnos. También es importante destacar, que de todos los docentes de la muestra, es el que más tiempo destinó a la realización de la secuencia.

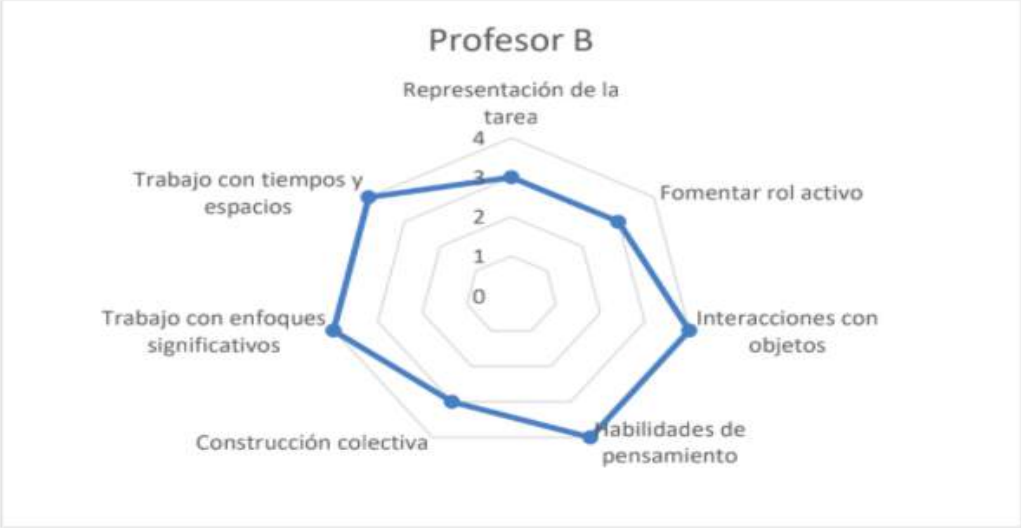


Figura 2. Fuente: elaboración propia

Por último, la figura 3 representa las observaciones en el último docente de la muestra (el docente C). Las observaciones realizadas sobre la implementación de su propuesta didáctica no muestra grandes variaciones con la de los otros docentes. Se observan algunas cuestiones referidas, en general, al tiempo dedicado a llevar adelante las actividades. El docente refiere que solo pudo dedicarle tiempos parciales al trabajo destinado a la investigación y desarrollo de las actividades debiendo, por calendario escolar, alternar con otros tratamientos de contenidos en forma paralela con el grupo. Este problema bastante común en la docencia, posiblemente se refleje en los menores valores en las actividades destinadas a la evaluación y representación de la tarea.

De todos modos, la potencialidad de lo actuado se observa en otros valores restantes de las categorías.

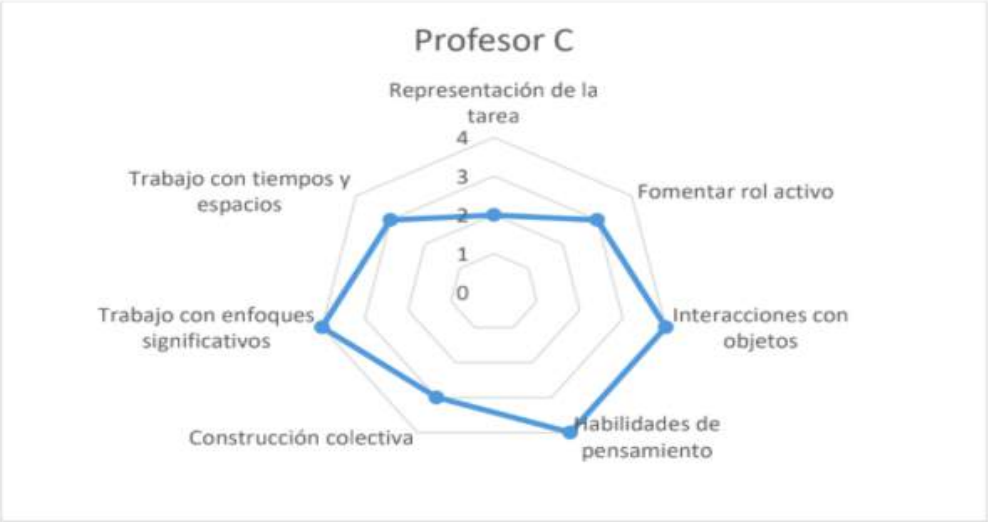


Figura 3. Fuente: elaboración propia

-Sobre el valor formativo para los docentes:

El análisis de las entrevistas iniciales y las de autoconfrontación realizadas a los docentes de la muestra, nos dicen en principio que éstos valoraron la propuesta en forma muy positiva, en términos generales. Se presenta a continuación un breve resumen de los aspectos más relevantes en relación a cada uno de los principales objetos de estudio de esta investigación.

Con relación a la secuencia didáctica propuesta, las reformulaciones que propusieron los docentes conservaron en gran medida el grado de apertura de la secuencia original, aun cuando se desempeñan en distintas materias, carreras y orientaciones. Los docentes implementaron las secuencias realizando algunas variaciones tanto en el objetivo específico (algunos centrados en contenidos disciplinares, y otros en habilidades cognitivas determinadas) como en el formato de trabajo (modificaciones a la secuenciación temporal, incorporación, eliminación o modificación de actividades previstas inicialmente, especialmente relacionadas con la integración de tecnología digital).

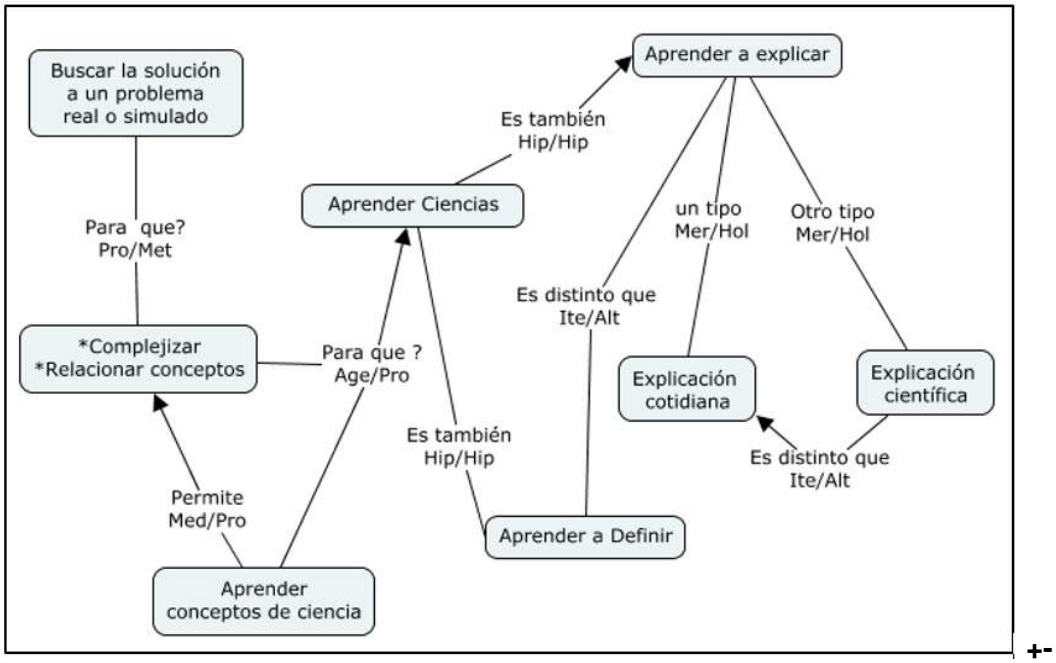
Se destaca la cantidad de cuestiones (no incluidas en principio por los docentes en la consigna de la actividad) que emergieron en la implementación. Esta situación podría ser considerada como muestra del importante grado de apropiación de los alumnos con respecto a la tarea encomendada. Estas cuestiones consideradas como emergentes de la propuesta original, dan la posibilidad al docente de multiplicar y articular las temáticas a trabajar, observándose que las mismas responden a muy distintos tipos de saberes de diversos campos disciplinares (matemáticos, de diseño, sociales, tecnológicos, didácticos, etc.). Consideramos que la cantidad y variedad de estos emergentes habla de numerosas interacciones posibles, y por consiguiente, de la potencialidad didáctica de la propuesta.

Un aspecto que resultó de gran interés para los participantes fue la posibilidad de trabajar “en” y “sobre” la práctica, numerosos aspectos didácticos, pedagógicos y disciplinares específicos. Compartir experiencias y reflexiones en las entrevistas de auto confrontación

(análisis de las prácticas) les permitió comprobar que muchas de las cuestiones y decisiones didácticas que tuvieron que tomar en la implementación de la secuencia, fueron muchas veces comunes a otros docentes de la muestra, y vinculadas a saberes profesionales que merecen ser explicitados y puestos en valor.

Otro resultado interesante, que aporta el análisis del discurso a la caracterización de la propuesta didáctica se relaciona con el estudio de los *circuitos dialógicos (*De Longhi, 2012). Consideramos que la interacción comunicativa reúne las características del tipo *Diálogo controlado con feedback,* con alguna tendencia hacia *Indagación dialógica orientada por el docente*, lo que pone de manifiesto un esquema de alta potencialidad comunicativa y didáctica.

El valor formativo de la propuesta analizada también puede verse reflejado en algunos de los resultados obtenidos por medio de las técnicas de análisis del discurso, como son los esquemas de patrón temático. Recordamos que estos esquemas se elaboran analizando el tipo de conectores que utiliza el docente en su discurso, y su función semántica, como por ejemplo relaciones nominales, taxonómicas, de transitividad, circunstanciales y lógicas. También se analizan las estrategias de diálogo y de monólogo que establece el profesor, para elaborar el patrón temático. En este caso separamos el patrón temático en "didáctico" y "disciplinar", y lo presentamos a continuación:



+-

Figura 4. Patrón temático didáctico de la clase. Fuente: elaboración propia

La figura 4 muestra el patrón temático didáctico de la clase. Presenta una gran riqueza y diversidad, y se articula con el patrón temático disciplinar. Los temas centrales abordados por el profesor se vinculan con la valoración didáctica de la resolución de problemas como estrategia de enseñanza, y las diversas habilidades cognitivas que permite desarrollar (aprender conceptos, a explicar, a definir, a relacionar, etc.). Consideramos que la

integración entre lo didáctico y lo disciplinar, y la complejidad alcanzada son posibilitadas por el tipo de secuencia didáctica planificada.

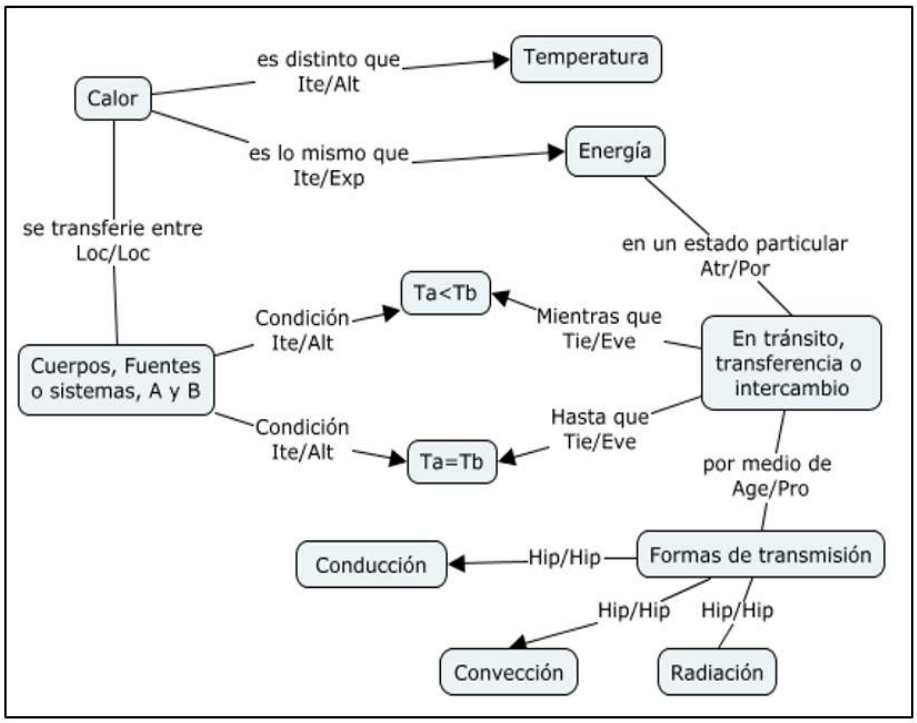


Figura 5. Patrón temático disciplinar de la clase. Fuente: elaboración propia

La figura 5 muestra el patrón temático disciplinar trabajado en la clase analizada, donde se advierten conceptos clásicos vinculados al estudio de la termodinámica (diferencia entre calor y temperatura, calor y energía térmica, formas de transmisión de energía térmica entre sistemas, etc.), organizados de una manera jerárquica y secuenciada según una lógica que pone de manifiesto los principales puntos de atención del profesor en la elaboración de su discurso.

El estudio de estos patrones temáticos se constituye como una herramienta formativa, permitiendo explicitar las estructuras semánticas que ponen en juego los docentes en los procesos de enseñanza, como por ejemplo con conceptos centrales, las relaciones jerárquicas, las lógicas subyacentes, etc., promoviendo un proceso reflexión sobre la práctica basado en elementos conceptuales sólidos, que van más allá de la simple opinión personal.

Un elemento emergente en este estudio, que señalamos como una dimensión digna de investigar, tiene que ver con las representaciones sociales de los docentes en relación a lo tecnológico en general y a las tecnologías digitales en particular. En el diagnóstico de nuestro trabajo señalábamos que las dificultades de integración curricular de tecnologías digitales estaban condicionadas en gran medida por factores externos al profesorado y, en menor medida, a cuestiones formativas. Sin embargo, valores simbólicos, representaciones sociales compartidas y creencias, verdaderamente podrían determinar el éxito o el fracaso de cualquier reforma curricular (Porlán y Rivero, 1998). La integración tecnológica digital, no

escapa de esta lógica y creemos que podría obstaculizarse o potenciarse por factores inconscientes y/o percepciones de sí mismo del profesorado.

En relación a la integración de saberes propios de las ciencias de la computación. La secuencia propuesta no estaba prevista inicialmente para la enseñanza de contenidos relacionados con la programación, sin embargo requirió de los docentes un trabajo previo para la implementación y calibración de un multitermómetro basado en Arduino.

En este sentido, los docentes de la muestra abordaron una concepción de programación secuencial (sin referencias explícitas a la programación orientada a objetos, los contenidos, muy restringidos, vinculados al uso de variables, uso de segmentos de código sin intervención, y un alcance de abordaje básico, implementado como parte de un proceso de alfabetización básica. Consideramos que, si bien se desarrolla un primer acercamiento al área de ciencias de la computación, se desaprovecharon posibilidades de avanzar, profundizar e interrelacionar saberes de interés y relevancia. Uno de los profesores (caso B) pudo avanzar en el análisis más detallado del código. Otros dos de los docentes del grupo se concentraron solamente en las ventajas técnicas y experimentales del instrumento, para medir temperaturas en varios en puntos simultáneamente, en función del tiempo.

La propuesta resultó formativa en este sentido, para los tres docentes de la muestra.

-Sobre los aprendizajes de los estudiantes:

En relación a los aprendizajes de los estudiantes, se valoran las vinculaciones de los contenidos disciplinares con problemáticas significativas, aunque resta profundizar el desarrollo de instrumentos de evaluación más adecuados al tipo de propuesta, que pudieran superar las formas y limitaciones de los dispositivos tradicionales de evaluación. Fueron valorados especialmente los espacios de diálogo en forma de coloquio, de lo que surgió, en algunos casos, la necesidad de abordar y profundizar el estudio de las formas y dinámicas de comunicación en el aula, el análisis de los discursos desde perspectivas formales, semánticas y retóricas, para propiciar y promover mejores aprendizajes.

En relación al análisis del discurso se indagaron los patrones temáticos de los estudiantes, al momento de presentar sus trabajos grupales, antes de la intervención discursiva del profesor. Resulta evidente una gran diferencia entre estos patrones temáticos iniciales de los estudiantes entre sí y con el profesor. Esta situación resulta adecuada considerando que el desarrollo de la clase (y las futuras clases dentro del curso) deberían promover un “acercamiento”, o construcción conjunta, entre el patrón temático presentado inicialmente por los grupos de estudiantes y el patrón temático presentado por el profesor.

Presentamos los conceptos o términos de interés de cada grupo, y sus vinculaciones semánticas principales:

Grupo 1. Enfoque “materiales, insolación y adaptación” para el diseño de la vivienda:

- La presentación se centra en las propiedades térmicas de los distintos materiales para la construcción, clasificados como buenos y malos conductores térmicos.

- Siguiendo esta línea argumental se considera la posibilidad de incluir el “vacío” como material, botellas o cubiertas con “vacío” o aire adentro.
- Orientación geográfica e insolación, y su impacto en el rendimiento térmico.
- Posibilidad de modificar el comportamiento térmico de la casa en distintas estaciones del año.
- Funciones a realizar dentro de la casa, necesidad de buena iluminación.

Grupo 2. Enfoque centrado en la “experiencia personal”:

- La presentación se centra en una serie de experiencias personales, en casas construidas de distinta forma y con distintos materiales, y los resultados obtenidos.
- Se argumenta en favor del ahorro de dinero, uso de materiales económicos, por ejemplo el adobe.
- Se concentra en el diseño de las aberturas y el sistema de calefacción.

Grupo 3. Enfoque “arquitectónico” del diseño de la vivienda (Int. 1,2,3,4,5,7):

- Materiales para la construcción (nombra varios: plástico, madera, barro, mármol, arena, residuos, adobe, pallets, agua, pintura, vidrio).
- Propiedades de los materiales (durabilidad, que no se pudran, precio, limpieza) (ninguno relacionado con el calor y la temperatura).
- Propiedades de la casa (duración, costo, comodidad, limpieza, privacidad) (relacionadas con la calidad de vida, como pautas de diseño arquitectónico, no tanto térmico).
- Elementos constructivos de una vivienda (piso, pared, techo, aberturas).
- Fuente de información : “Internet”.

Grupo 4. Enfoque “térmico” del diseño de la vivienda (Int. 13,15,16,18, 20):

- Procesos de intercambio de calor (radiación (del sol), convección (circulación de aire interior-exterior), conducción (materiales aislantes o conductores térmicos).
- Otros procesos complejos: Efecto invernadero, funcionamiento del muro Trombe.
- Influencia de la orientación geográfica en el funcionamiento térmico de la casa (estaciones del año, inclinación de la radiación solar) .

Otro aspecto relacionado al aprendizaje de los estudiantes se vincula a la conceptualización sobre el desarrollo científico, desde una perspectiva falsacionista, en la que la forma de estructurar la solución a un problema concreto, puede servir de base para su adaptación a otro problema de naturaleza similar. En este sentido, la contextualización de la enseñanza en relación a problemas complejos, permite el trabajo sobre la polisemia de significados, que suele aparecer habitualmente como problema en la enseñanza de las ciencias, permitiendo avanzar hacia el desarrollo de habilidades cognitivas de orden superior.

En relación a los aprendizajes sobre cuestiones tecnológicas, se observó que en las etapas de diseño se incorporaron diversas estrategias constructivas o hipótesis de construcción para la resolver el problema base que implicaban la utilización de diversas tecnologías analógicas (por ejemplo: muro Trombe, ladrillos con aire estanco, ubicación y tipo de forestación, etc.). Algunas de estas, fueron incluidas en los proyectos a partir del

descubrimiento de estas posibilidades en tutoriales o fuentes de información en línea. Este hecho evidencia dos cuestiones de fondo, la primera es la percepción de sí mismo de los estudiantes como idóneos en la explicación, comprensión de los modelos que se imbrican y posible inclusión de estas soluciones tecnológicas en sus proyectos modélicos y aún en construcciones reales. La segunda cuestión es el hecho que ningún grupo hizo mención a la inclusión de tecnologías digitales en sus proyectos (por ejemplo, diseño CAD, controladores de parámetros que mejoren la habitabilidad, relevamiento automatizado de datos del modelo en tiempo real, etc.). La aplicación de tecnologías digitales están implícitas o naturalizadas en la utilización de motores de búsqueda a través de sus celulares, escritura de informes, etc. Esta relación o vínculo con las tecnologías digitales visibiliza una naturalización de estas estrategias de búsqueda de información, a la vez que deja en evidencia la precariedad de las enseñanzas y aprendizajes en tecnologías digitales que el sistema educativo ha provisto como alfabetización digital en los trayectos de formación de los distintos niveles de escolarización.

En relación a contenidos vinculados con la programación, el abordaje fue, en el caso de estudio, de tipo introductorio y general. Para la mayoría de los estudiantes se trató de un primer contacto, en su trayectoria formativa, con un entorno de programación (el IDE de Arduino). En este sentido señalamos algunos aspectos que abordados, como reconocer la existencia de un código que debe ser compilado e inyectado en el procesador, identificar las estructuras básicas del programa utilizado (tomado de tutoriales de difusión libre), realizar la identificación y control de variables de interés en el modelo (temperaturas y tiempos) y sus vínculos con el tipo de variable y su almacenamiento y con los dispositivos electrónicos que las generan, la identificación y modificación de parámetros de interés (para lectura, registro y presentación de los datos). No se realizó un análisis sobre el tipo de entorno o lenguaje de programación utilizado, ya que se trató de una primera aproximación a la programación, vinculada al análisis, configuración y adaptación de programas desarrollados por otros autores, en un contexto determinado por la resolución de un problema de interés para los estudiantes.

Conclusiones

Las secuencias de enseñanza analizadas se pueden caracterizar como de “alta potencialidad didáctica”, en relación al tipo de propuesta (problema abierto contextualizado, generación de emergentes, diversidad de respuestas de los distintos grupos, formas de agrupamiento, desarrollo en tiempos razonables, etc.) y también en relación a la riqueza de los patrones temáticos que posibilitan (tanto disciplinares como didácticos), y al tipo de circuito comunicacional que generan (indagación dialógica orientada por el docente).

Existen ciertos aspectos que se podrían mejorar o intensificar en el desarrollo de propuestas didácticas en el marco STEM, como las estudiadas en este caso, podemos señalar la posibilidad de un tratamiento más valorado de contenidos de disciplinas emergentes en el proceso de resolución por medio del trabajo articulado con profesores de cada disciplina, que podrían planificarse al diseñar las secuencias. En particular, la intervención de

profesores del área de las ciencias de la computación podría caracterizar y jerarquizar los aportes de estas disciplinas a la resolución del problema planteado en la secuencia, que resulten pertinentes.

En relación al enfoque metodológico adoptado en esta investigación, consideramos que tanto las técnicas del análisis de la práctica como las del análisis del discurso resultan pertinentes y productivas para el estudio de la potencialidad didáctica de propuestas STEM, y se presentan como instrumentos adecuados para la profundización y desarrollo del estudio didáctico, especialmente en relación a la participación de docentes del área de ciencias de la computación en este tipo de proyectos de integración disciplinar en el área STEM.

En particular el estudio realizado pone de manifiesto la necesidad de elaborar herramientas de investigación específicas para reconocer y caracterizar las representaciones de los docentes y estudiantes participantes sobre la tecnología, y su influencia en el diseño de propuestas didácticas de integración. En este sentido, el estudio de las representaciones sociales de los diferentes colectivos podría ayudarnos a conocer lo aspectos simbólico para evaluar la incidencia de estos sistemas simbólicos en la implementación de estrategias de integración tecnológica. Esta línea de trabajo se encuentra actualmente en desarrollo por los autores.

Otras conclusiones que surgen del presente estudio y de los intercambios mantenidos en las JADIPRO 2019, se relacionan con orientaciones a tener en cuenta en los procesos de diseño y planificación de dispositivos de formación docente inicial y continua, de materias vinculadas al área STE. En primer lugar, la importancia de sostener y poner en valor la jerarquía de las ciencias de la computación, evitando enfoques reduccionistas o triviales, bajo una supuesta multidisciplinariedad, difícilmente alcanzable en ámbitos profesionales o escolares. Una forma de viabilizar esta propuesta es trabajando en forma conjunta, desde la planificación didáctica, con profesores especialistas de cada una de las disciplinas involucradas, con un enfoque abierto y compartido, para abordar problemas adecuados y con suficiente grado de complejidad y significatividad.

Por otra parte, el abordaje de problemas complejos, de relevancia social, de forma abierta, pone de manifiesto la importancia del trabajo con el desarrollo de criterios para valorar la información circulante en los medios y redes (*fake news*, información sin validar, fuentes dudosas, manipulación en medios, etc.), y la reflexión sobre la imagen de ciencia y tecnología que se construye, desde la perspectiva de “naturaleza de las ciencias”. Esta perspectiva de abordaje de los problemas, que va más allá de una simple resolución técnica, habilita y demanda una visión en perspectiva histórica sobre las distintas tendencias en la historia de las ciencias de la computación, sus vínculos con los sistemas productivos, políticos y académicos, y su relación con su enseñanza escolar, complejizando ampliando los marcos de referencia.

También resultan aspectos de interés los aportes que podrían hacer las didácticas específicas más consolidadas (como las de la matemática o las ciencias naturales) al desarrollo de la didáctica específica de las ciencias de la computación, señalando y considerando algunas similitudes y diferencias epistemológicas específicas de cada campo disciplinar.

En síntesis, creemos que un proceso compartido -en el que, partiendo de propuestas didácticas previamente diseñadas, que involucran tecnología accesible y con un grado de apertura importante, que permite su análisis y adaptación en función de los problemas y estilos del propio docente que debe implementarla, y que finalmente incluye una reflexión docente sobre la propia práctica, andamiada por una metodología que involucra una mirada conjunta de lo realizado en el aula-, constituye un espacio de aprendizaje que al mismo tiempo, genera insumos para el diseño y mejora de propuestas de alta potencialidad didáctica para la enseñanza de la Física, y puede ser recomendada como una estrategia de enseñanza disciplinar y de formación docente en otras disciplinas del área STEM, en particular las vinculadas a las ciencias de la computación.

De esta manera, consideramos adecuadamente cumplidos los objetivos de esta investigación, caracterizando y analizando el dispositivo didáctico propuesto, que se muestra como una herramienta útil para la formación docente, inicial y continua en disciplinas del área STEM.

Agradecimientos

A la Universidad Pedagógica Nacional por el apoyo otorgado para la realización de la investigación de la que este trabajo forma parte mediante el subsidio correspondiente a la resolución CS7/2018.

A la comunidad educativa del ISFDN°99 “Unidad académica Almafuerte” por otorgarnos el espacio y el apoyo para realizar el trabajo de campo en esta institución.

Bibliografía

Adell, J. S., Llopis, M. A. N., Esteve, M. F. M., y Valdeolivas, N. M. G. (2019). El debate sobre el pensamiento computacional en educación. En *RIED, Revista Iberoamericana de Educación a Distancia,* 22(1), pp. 171-186. doi: http://dx.doi.org/10.5944/ried.22.1.22303

Adúriz-Bravo, A. (2010). Hacia una didáctica de las ciencias experimentales basada en modelos. CiDd. II Congrés Internacional de Didactiques. 2010.

Adúriz-Bravo, Agustín. (2018). Argumentación basada en modelos desde la perspectiva de la epistemología y la historia de la ciencia. *En Revista Tecné, Episteme y Didaxis*. Año 2018. Número Extraordinario.

Araya, R. (2016). STEM y modelamiento matemático. *En Rev. Elec. Diál.Educativos.* ISSN: 0718-1310.

Bardin, L. (1991). *Análisis de contenido*. Madrid: Ediciones Akal.

Bordignon, F. e Iglesias, A. (2015) *Diseño y construcción de objetos interactivos digitales. Prácticas con Arduino.* CABA: UNIPE Editorial Universitaria.

Brown, J. (2012). The current status of STEM education research. En *Journal of STEM Education: Innovations and Research*, *13*(5), 7.

Bybee, Rodger W. (2010). Advancing STEM Education: A 2020 Vision. En *Technology and Engineering Teacher,* 70 (1), pp. 30-35.

CFE Consejo Federal de Educación, (2017). Marco de Organización de los Aprendizajes para la Educación Obligatoria Argentina (MOA). Resolución CFE 330/2017.

CFE Consejo Federal de Educación (2018) Núcleos de Aprendizaje Prioritarios para educación Digital, Programación y Robótica. Resolución CFE 343/18.

De Longhi, A. et al. (2012). La interacción discursiva en el aula de ciencias naturales. Un análisis didáctico a través de circuitos discursivos. *En Revista Eureka sobre la enseñanza y divulgación de las ciencias.* 9(2), pp. 198-195.

Di Giácomo, J.P. (1987) Teoría y métodos de análisis de las representaciones sociales. En *Pensamiento, individuo y sociedad: cognición y representación social.*

Furci, V., Trinidad O, y Peretti, L. (2018) Actividades experimentales abiertas mediadas por tecnología Arduino como propuesta de formación docente en Física. En *Revista de Enseñanza de la Física*. Vol. 30, No. Extra, Nov. 2018, pp. 83-89.

Furci, V., Gonzalez A., Trinidad O., Peretti L. (2019). Análisis discursivo de la potencialidad didáctica de una propuesta STEM. Congreso LASERA 2019. Guanajuato. México. En *Revista Latinoamericana de enseñanza de las ciencias*. (en prensa).

Galagovsky, L. y Adúriz-Bravo, A. (2001). Modelos y analogías en la enseñanza de las ciencias naturales. El concepto de modelo didáctico analógico. En *Enseñanza de las Ciencias*, *19*(2), pp. 231-242.

García, Y., González, D. S. R., y Oviedo, F. B. (2017). Actividades STEM en la formación inicial de profesores. En *Diálogos educativos*, (33), pp. 35-46.

Gutiérrez,A.; Furci, V. ; Trinidad, O.; Pedrol, H. (2018). STEM en la formación docente inicial: el laboratorio de ciencias como espacio de innovación. Vol 4, pag 25. Libro: *Enseñanza y Aprendizaje de las Ciencias en Debate*. CIEDUC, 2018. Montevideo. Uruguay.

King, D. (2012). New perspectives on context-based chemistry education: Using a dialectical sociocultural approach to view teaching and learning. *Studies in Science Education*, 48(1), pp. 51-87.

Kortland, J. (2007). Context-based science curricula: Exploring the didactical friction between context and science content. En ESERA 2007 Conference. Malmo, Sweden.

Litwin, E. (2008). *El oficio de enseñar.* Buenos Aires, Argentina: Paidos.

Moraga Toledo, S; Espinet Blanch, M. y Merino Rubilar, C. (2019) El contexto en la enseñanza de la Química: Análisis de secuencias de enseñanza y aprendizaje diseñadas por profesores de ciencias de secundaria en formación inicial. En *Revista Eureka sobre la Enseñanza y Divulgación de las Ciencias.* (16) 1- 14.

Ortiz-Revilla, J., Adúriz-Bravo, A. y Greca, I.M. A Framework for Epistemological Discussion on Integrated STEM Education. En *Sci&Educ* (2020). https://doi.org/10.1007/s11191-020-00131-9

Park, S. y Oliver, S. (2007). Revisiting the conceptualisation of pedagogical content Knowledge: PCK as a conceptual tool to understand teachers as professionals. Res. En *Sci. Educ,* 38 (3), pp. 261-284.

Peretti, L., Furci, V., y Trinidad, O. (2019). Algunas reflexiones filosóficas y didácticas en torno a propuestas STEM como contexto de enseñanza de las ciencias naturales: Potencialidades y riesgos de un movimiento hegemónico. En *1º Congreso Internacional de Ciencias Humanas-Humanidades entre pasado y futuro*. Escuela de Humanidades, Universidad Nacional de San Martín.

Peretti, L.; Furci, V. y Trinidad, O. (2019). Algunas reflexiones filosófico-didácticas en torno a propuestas STEM como contexto de enseñanza de las ciencias naturales y su inclusión en los currículos. Presentado en el congreso LASERA 2019. Guanajuato. México. En Revista Latinoamericana de enseñanza de las ciencias. (en prensa).

Piña Osorio, J; y Cuevas Cajiga, Yazmin (2004) La teoría de las representaciones sociales. Su uso en la investigación educativa en México. En *Perfiles educativos* vol. 26. N° 105 México.

Porlán, R., Rivero, A. y Martín del Pozo, R. (1998) Conocimiento profesional y epistemología de los profesores II: Estudios empíricos y conclusiones, En *Enseñanza de las Ciencias,*16:2, pp. 271-288.

Rickenmann, R. (2007). Metodologías clínicas de investigación en didácticas y formación del profesorado: un estudio de los dispositivos de formación en alternancia. En *Revista Científica Sao Pablo,* vol 9, nº 2 jul/dic 2007, pp. 435-463.

Fundación Sadosky (2013). CC–2016 Una propuesta para refundar la enseñanza de la computación en las escuelas Argentinas. Disponible en https://cutt.ly/xp4MIud

Szpiniak, A. F., y Rojo, G. A. (2006). Enseñanza de la programación. En *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, (1), 8-p.

Trinidad O., Furci V., Bonet A., Peretti L., y Dicosmo. C. (2019).Formación docente en contexto STEM: Potencialidad didáctica de actividades experimentales abiertas mediadas por tecnología en la enseñanza de la Física. Presentado en el congreso LASERA 2019. Guanajuato. México. En *Revista Latinoamericana de enseñanza de las ciencias.* (en prensa).

Trinidad, O., Furci, V., y Peretti, L. (2019) Formación docente en contexto STEM: actividades experimentales abiertas mediadas por tecnología Arduino en la enseñanza de la Física. En *Revista de Enseñanza de la Física*, *31*, 707-714.

Zapata, J. (2016). Contexto en la enseñanza de las ciencias: análisis al contexto en la enseñanza de la Fisica. En *Góndola, Enseñanza y Aprendizaje de las Ciencias*. 11(2), 193-211.

Zoller, U., Blonder, R., Finlayson, o. E., Bogner, F., y Anne, I. (2013). Research-based coherent science teaching-assessment-learning to think for global sustainability. Strand 2 learning science: cognitive, affective, and social aspects, 170.

Anexo 1

Tabla con variables y categorías de análisis de la dimensión "potencialidad didáctica"

TABLA I. Variables y categorías de análisis de la dimensión "Potencialidad Didáctica".

Variables	Categorías de Análisis
V1 Representación de la tarea	**-4 Espacio explícito:** Momentos explícitos en donde docente y alumnos expresan representaciones de la tarea en función de la evaluación. **-3 Espacio parcial explícito:** Momento de representación de una faceta particular de la tarea en función de la resolución de alguna actividad. **-2 Espacio emergente:** Espacio que se crea por problemas emergentes ocasionales. **-1 No existe ese espacio:** Los criterios de resolución de la tarea son implícitos y son potestad del profesor
V2 Fomentan rol activo	**-4 Formular problemas:** La actividad planteada, se encuentra en función de problemas de interés, planteados por los alumnos. **-3 Tomar problemas emergentes**: La actividad planteada por el profesor, puede variar en función de problemas emergentes planteado por los alumnos. **-2 Resolver problemas:** La actividad principal consiste en resolver problemas planteados por el profesor con métodos propios **-1 Resolver:** La actividad se centra en seguir protocolos planteados por el profesor siguiendo pasos establecidos
V3 Interacciones con objetos	**-4 Investigar**: Construir un problema y resolverlo, diseñando procedimiento, dispositivos y validando las respuestas encontradas. Ej1: diseñar un dispositivo para medir la contaminación en el aire. Ej2:crear un programa que haga sonar notas según como muevo mi mano) **-3 Hackear un objeto o dispositivo**. A partir de un dispositivo conocido, modificarlo para nuevo uso o utilización en otro contexto Ej1: Utilizar el láser de un lector de CD y usarlo para experiencias de óptica Ej2: Utilizar un multímetro para el diseño y construcción de un dispositivo que estime humedad de suelos **-2 Experimentar**: Diseñar un procedimiento, controlando variables, para comprobar propiedades conocidas de un fenómeno u objeto. Ej1: Diseñar un procedimiento para comprobar la ley de Ohm. E2: Diseñar un procedimiento para evaluar el rango de trabajo de un dispositivo. **-1 Seguir protocolos:** Realizar una cierta actividad experimentar encomendada a partir de un protocolo y materiales u objetos dados. Ej1: Comprobar experimentalmente las temperaturas de cambio de estado del agua, utilizando materiales y procedimientos dados por el docente. Ej2: Construir un cierto dispositivo siguiendo un tutorial.
V4 Habilidades de pensamiento	**-4 Mayormente HOCS**: En la resolución de actividades se fomenta mayormente habilidades como: argumentar, decidir, diseñar, relacionar, evaluar, etc. **-3 Habilidades HOCS y LOCS:** En la resolución de actividades se observan habilidades de pensamiento de orden superior e inferior integradas. **-2 Mayormente LOCS**: La resolución de las actividades, demandan mayormente habilidades de pensamiento de orden inferior: observar, describir, identificar, aplicar, etc. **-1 Habilidades técnicas:** La actividades están centradas en "el hacer" no necesariamente reflexivo: copiar, manipular, medir, utilizar artefactos o dispositivos de acuerdo a pautas establecidas siguiendo un protocolo.
V5 Construcción colectiva	**-4 Formato de comunicación pública:** En la actividad presentada, se requiere en forma explícita, formatos de comunicación oral y escrita con términos técnicos y citas bibliográficas adecuadas, producto de trabajos previos de producción compartida **-3 Formato de comunicación interna:** En la actividad se fomentan formatos de comunicación fluida, en donde se expresan los aprendizajes logrados, aunque también los errores y dudas valorando éstos, como avances en la producción y el aprendizaje **-2 Sin formato establecido:** La actividad contempla informes final sobre las tareas realizadas, aunque no establece formatos para la misma **-1 Sin comunicación:** La actividad realizada no requiere informes
V6 Trabajo con enfoques significativo	**-4 STEAM**: Las actividades realizadas contemplan relaciones entre contenidos STEAM **-3 Enfoques CTSA:** Las actividades realizadas contemplan claras relaciones entre contenidos CTSA **-2 Relaciones ocasionales**: Las actividades realizadas contemplan relaciones ocasionales entre contenidos de otras disciplinas o problemas cotidianos **-1 Campo de la Física**: Las actividades realizadas solo presentan trabajos con contenidos de la disciplina
V7 Trabajo en tiempos y espacios adecuados	**-4 Ambiente escolar**, materiales, tiempos y espacios se corresponden con la propuesta didáctica **-3 El ambiente es adecuado**, pero tiempos y espacios son responsabilidad **compartida** de alumnos y docente **-2 Tiempos y espacios** son mayormente **aportados por los alumnos** **-1 Tiempos y espacios** son una **dificultad** para la producción de los alumnos

Robótica educativa con software libre y hardware de especificaciones abiertas para enseñanza de programación

Diego Letzen, Valentín Basel, Alba Massolo, Federico Ferrero

Resumen: El presente trabajo intenta dar cuenta de parte del trabajo desarrollado por los autores en la elaboración de una propuesta para la enseñanza de la programación en escuelas secundarias de la provincia de Córdoba. En el transcurso del proyecto de investigación, se evaluaron algunos de los paquetes de robótica educativa disponibles en Argentina a la fecha y se discutieron y diseñaron una serie de robots educativos basados en el hardware de desarrollo ICARO (placa NP07 con microcontroladores PIC 18F4550) diseñados con la finalidad de proponer intervenciones para la enseñanza de programación.

Palabras claves: Robótica educativa, software libre, hardware libre.

Introducción

A partir de las discusiones llevadas a cabo en el grupo de investigación se decidió seleccionar, para el desarrollo del robot, diferentes categorías de análisis que permitan dar cuenta de las distintas alternativas de desarrollo que actualmente se consiguen en el mercado, teniendo en cuenta las discusiones hechas sobre el uso de recursos educativos abiertos (Z. González & HernándezII, s. f.), software libre, hardware de especificaciones libres, soberanía tecnológica (Haché, 2014) y didáctica de la programación. Teniendo en cuenta esas dimensiones de análisis, se eligieron las siguientes categorías para seleccionar el software y hardware que usaría el equipo de investigación:

• Que sea desarrollado con software libre.
• Que las especificaciones del diseño de hardware sean libres.
• Que pueda ser fabricado de forma hogareña (DiY).
• Bajo costo de adquisición / fabricación.
• Buena documentación.
• Multiplataforma.
• Que tanto el hardware como el software permitan proyectar en complejidad de trabajo (sin encerrar en una "caja negra" partes del software y el hardware).
• La cantidad de propuestas educativas que pueden ser vinculadas al paquete.
• Es un proyecto recomendado para trabajar en el aula, en grupos.
• Es un proyecto recomendado para trabajar de forma individual.
• Los conocimientos necesarios para poder implementar su uso.
• Las comunidades de desarrollo de hardware y software que puedan dar soporte al proyecto.

Seleccionadas las categorías de análisis, se procedió a catalogar distintos proyectos comerciales y comunitarios que podrían servir para el desarrollo de la investigación. La necesidad de plantear un proyecto que se posicione desde una perspectiva de soberanía tecnológica, llevó a tomar como criterio principal el uso de software y hardware libre.

Software libre y hardware de especificaciones libres

El software libre es un movimiento que comenzó en el año 1983 cuando Richard Stallman anunció el proyecto GNU (Stallman, 2007) en contraposición a la aparición de monopolios artificiales en el desarrollo de software (Busaniche y otros, 2007). Se podría decir que la meta del movimiento fue dar libertad a los usuarios de programas de computadoras reemplazando el software con términos de licencias restrictivas (software privativo) por una alternativa libre que permitiera a los usuarios y desarrolladores de software contar con al menos cuatro libertades con respecto al uso del mismo, a saber:

• Libertad 0: la libertad de usar el programa, con cualquier propósito (uso).
• Libertad 1: la libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a las propias necesidades (estudio).

• Libertad 2: la libertad de distribuir copias del programa, con lo cual se puede ayudar a otros usuarios (distribución).
• Libertad 3: la libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie (mejora).

Por su parte, el hardware de especificaciones abiertas busca llevar el concepto del software libre (la libertad de usar, estudiar, distribuir o mejorar el software) al diseño de componentes físicos (Olivencia & Martínez, 2015), especificando una licencia que permite distribuir planos y código fuente de desarrollos de PCBs (Printed Circuit Board, por sus siglas en inglés) y hardware electrónico. Asimismo, se considera que un diseño de circuito (esquemático, diseño de PCB y archivos GERBER) debe ser desarrollado con software libre y usando formatos abiertos.

Se tomó una clasificación del hardware en función de su diseño y el software empleado para su creación (González, González, & Gómez-Arribas, 2003), definiendo una clasificación primaria de los tres tipos de archivos necesarios para la fabricación de un PCB, el esquemático, el archivo de pcb y el archivo GERBER. En función de esa clasificación, se puede separar al hardware en tres tipos:

• (P): Software de diseño propietario.
• (L): Software de diseño libre.
• (M): Software de diseño propietario pero multiplataforma (funciona en sistemas operativos libres y también en propietarios).

Si tenemos en cuenta los 3 planos electrónicos necesarios para fabricar un PCB podemos ver que necesitamos tener el archivo esquemático, el archivo de PCB y el fichero de fabricación. Por tanto, los tres archivos de construcción para el circuito electrónico pueden ser de clasificación como enteramente libres (tipo LLL) o completamente propietarios (PPX), y todas sus combinaciones posibles. Además, desde la perspectiva de la soberanía tecnológica (Haché, 2014), se decidió que la tecnología elegida para la construcción del hardware robot, fuera de fácil acceso, bajo costo y con materiales sencillos de trabajar sin necesidades de herramientas especiales.

Clasificando hardware

Para la elección del hardware de control se analizaron las dimensiones propuestas y sobre eso se armó una tabla de doble entrada que pusieran en comparación las distintas soluciones propuestas por el grupo.

Para clasificar el hardware que se podía obtener para la investigación se utilizó una métrica de 3 valores que permitieron seleccionar entre:

• Valor 0: No aplica al esquema de clasificación ponderado.

• Valor 1: Algunas partes del proyecto pueden ser consideradas en la categoría seleccionada.
• Valor 2: Aplica completamente a la categoría seleccionada.

	Lego mindstorms ev3	Robotgroup robot N6	Kit Arduino Electr. básica	Proyecto ICARO	Makeblock ultimate kit	Mis ladrillos probots R502
desarrollado con software libre.	0	1	2	2	1	0
Diseñado con Hardware libre	0	1	2	2	1	0
Capacidad Do It Yourself (DIY)	0	0	1	2	0	0
Bajo costo de adquisición / fabricación.	0	1	2	2	0	1
Documentación	2	1	2	1	1	1
Multiplataforma	0	0	2	2	1	0
Proyección de complejidad de trabajo (eliminar “cajas negras”)	1	1	2	2	1	0
propuestas educativas	2	1	2	1	1	1
Bajo Conocimientos necesarios para poder implementar su uso	2	1	1	1	1	2
comunidades de desarrollo de hardware y software	1	0	2	2	0	0
TOTAL	8	7	18	17	7	5

Tabla comparativa.

La confección de una tabla de doble entrada demostró rápidamente que los 2 únicos proyectos que cumplimentaban con la mayor cantidad de requerimientos eran el uso de robots basados en ARDUINO o el uso de hardware del proyecto ICARO (ver tabla comparativa).

Finalmente, se decidió trabajar usando el hardware y software del proyecto ICARO, dado que siendo la evaluación muy similar, en el caso de ICARO se tenía acceso a la comunidad de desarrollo del software, y la posibilidad de adecuar la electrónica y el software de control en función de las necesidades del grupo.

Tribot

En principio se optó por un esquema de robot tipo TRIBOT, con dos motores de corriente continua, 6 pilas AA, un sensor de ultrasonido (hc-sr04), un sensor infrarrojo (CNY70) y dos micro switch del tipo “final de carrera” como sensores digitales de contacto, como se puede ver en la figura 1. Para su programación se utilizó el software ICARO, creando el firmware de trabajo usando lenguaje ANSI C, con el compilador libre SDCC (Versión 3.7.0) y las bibliotecas del proyecto PINGUINO (bootloader v4).

Este robot TRIBOT (Zabala, 2007) se programa y carga su firmware mediante cable USB, siendo una mecánica de trabajo muy parecida a los esquemas propuestos por otros kits comerciales o diseñados por comunidades.

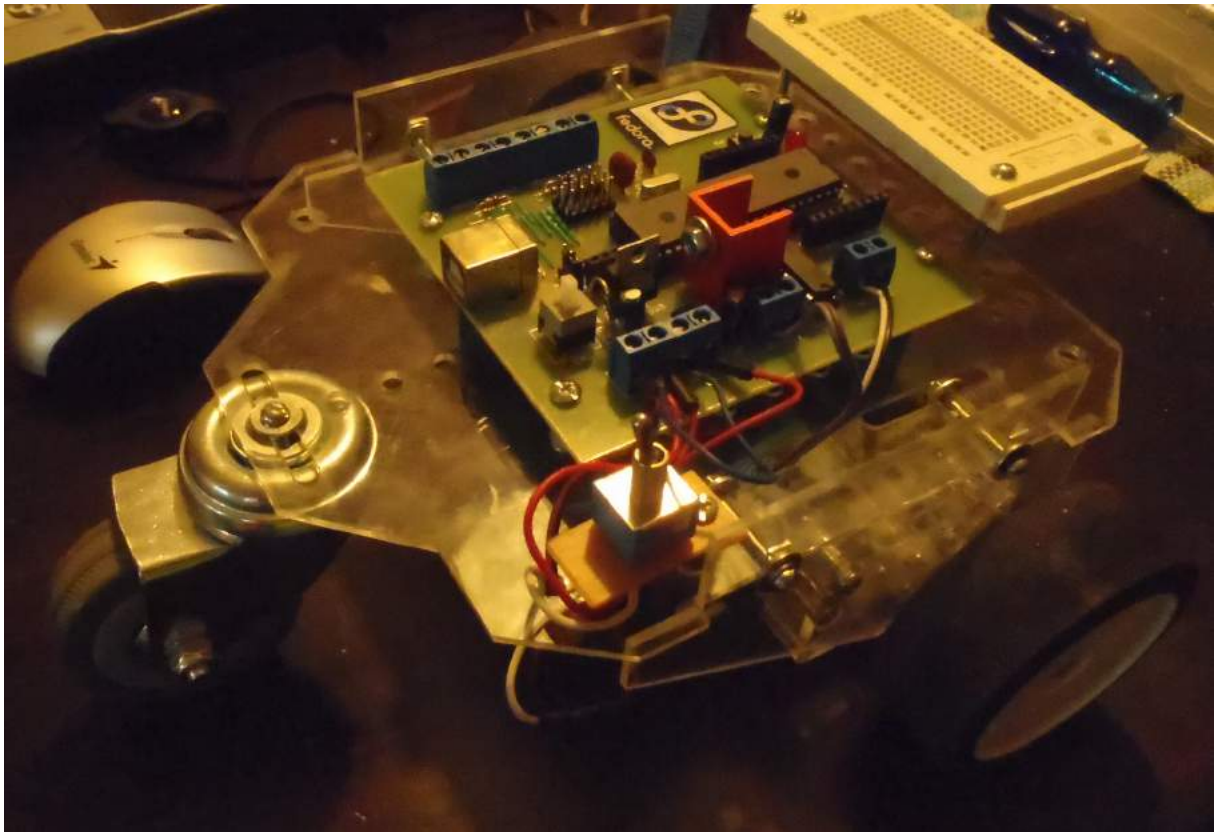


Figura 1. Prototipo de robot tipo TRIBOT.

La decisión de usar microcontroladores PICs en vez de hardware basado en ARDUINO (ATmega328), se debió principalmente a la ventaja de poder contar con un microcontrolador en formato THT (Through-Hole Technology) que tuviera una interface USB 2.0 por hardware y que fuera fácil de adquirir en pequeñas cantidades para prototipado. El formato THT permite trabajar con integrados grandes que se pueden manipular y soldar con un soldador de estaño hogareño y sin herramientas especiales. El prototipo funcionó satisfactoriamente pero con algunas limitaciones relacionadas con el uso del espacio en situaciones de aprendizaje grupal formal y la logística implicada en la dependencia de pilas para el funcionamiento. La observación que más peso tuvo a la hora de dejar de lado este tipo de

robots fue el requerimientos que los robots considerados pudieran dibujar con más o menos precisión para poder retomar algunos de los ejercicios propuestos por Papert y el sistema LOGO (Papert, 1980), como piso de la experiencia. El robot TRIBOT se mostró ineficiente en ese cometido, y por lo tanto se tomó la decisión de experimentar con plataformas fijas tipo PLOTTER o robots de tipo SCARA, que fueran lo más baratos y fáciles de fabricar y emplear en un aula de escuela secundaria de una escuela pública argentina.

Robot SCARA

En paralelo a las pruebas que se hicieron sobre el robot tribot, se tomó la decisión de fabricar un robot de tipo SCARA (acrónimo que responde por sus siglas en inglés a *Selective Compliant Assembly Robot Arm*) que permitiría trabajar con una plataforma conectada a la computadora y controlarlo mediante programación en PYTHON.

Los robots de tipo SCARA tienen una cinemática inversa del movimiento sencilla de calcular, y tienen un margen amplio de movimientos, aunque no tienen las capacidades de un brazo robot de 6 grados de libertad (Reyes Cortés, 2011), son más fáciles de fabricar y tienen mejor relación de fuerza. Además, los robots SCARA son generalmente más rápidos y sencillos que los sistemas comparables de robots cartesianos.

Para su fabricación, se eligió continuar con los componentes electrónicos, eléctricos y mecánicos de fácil adquisición que se usaron para el robot tribot, aunque los motores paso a paso 28BYJ-48 se mostraron demasiado lentos y de escasa fuerza como para poder mover los ejes del robot, por lo tanto se eligió trabajar con servomotores de aeromodelismo.

Los servomotores de aeromodelismo, se presentan como una alternativa interesante por su bajo costo, fácil control a través de PWM (siglas en inglés de *pulse-width modulation*), potencia y velocidad, además de poseer toda la electrónica de control dentro del mismo servomotor, lo que reduce mucho el desarrollo de electrónica de control.

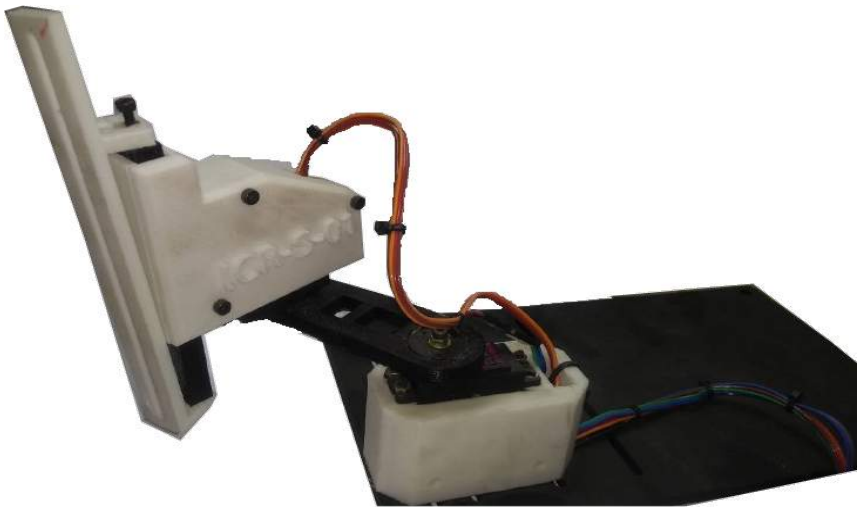


Figura 2. Robot tipo SCARA.

El robot SCARA mínimo y funcional necesita tener 3 servomotores, generalmente dos servos “grandes” (con un torque de 3.7 kg-cm) que controlan el “hombro” y “codo” del brazo robot, y un micro servo 9g para el actuador.

En las pruebas realizadas, los servomotores resultaron tener poca precisión como para dibujar sobre una superficie, aunque era muy veloces y con bastante potencia como para mover piezas pequeñas y trabajar con un sistema más parecido al uso industrial que generalmente se da con estas plataformas, por lo que se decidió considerar una nueva alternativa para solucionar estos inconvenientes.

V-plotter

Si bien los robots SCARA pueden hacer trazos, al usar servo motores de poca precisión, no se obtuvieron resultados que justificaran su uso para hacer dibujos, y por lo tanto se decidió desarrollar una plataforma basado en un sistema V-plotter.

Estos son uno de los modelos más sencillos de plotters para fabricar, en los que el cabezal de dibujo cuelga atado a dos cables que se enrollan en dos poleas a cada lado de la superficie de trabajo. Estas poleas son controladas por motores “paso a paso” que enrollando y desenrollando su cable, mueven el cabezal en los ejes X/Y.

Este modelo de plataforma robot se presenta como una alternativa fácil de fabricar respecto a plotters de tipo cartesiano (impresoras 3d por ejemplo), o robots de movimientos polares tipo SCARA, dado que solo tienen un cabezal colgando de dos poleas, sobre una plataforma vertical.

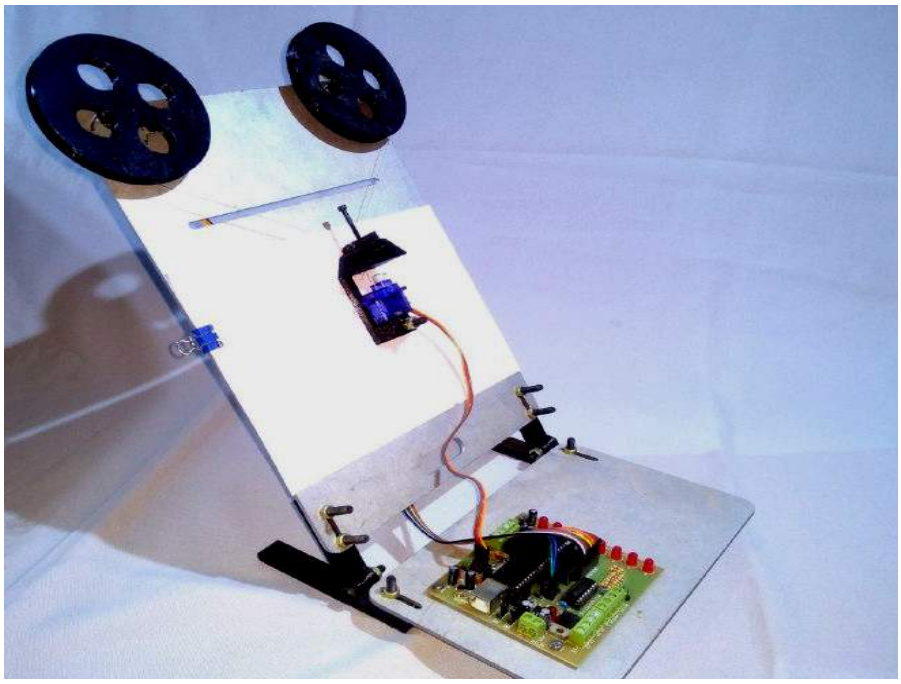


Figura 3. Plataforma Vplotter final.

Para el desarrollo del V-PLOTTER se decidió trabajar con 2 motores PAP (28BYJ-48) de bajo coste y micro servos 9g en el cabezal para poder levantar el lápiz y mover el cabezal sin dibujar. La particularidad de los motores 28BYJ-48 es su bajo torque y escasa velocidad de movimientos, sin embargo, al ser motores de 4 pasos (step) u 8 medio pasos (half steps) con caja reductora 1/64, dan como resultado que para hacer una vuelta entera, el motor debe recibir 512 impulsos (cada "paso" del motor equivale a 0,703125 grados) dando un buen margen de precisión de movimiento para la plataforma robot. Para el control de la plataforma se optó por trabajar con bibliotecas escritas en Python 2.7 que permitió el desarrollo de una API (*Application programming interface*) que implementa los algoritmos necesarios para poder controlar el movimiento de de los motores, y pudiera servir como un paralelismo entre el lenguaje LOGO y sus primitivas (adelante, atrás, izquierda, derecha), para permitir trabajar desde Python pero con instrucciones parecidas a LOGO.

Mecánica

Al ser una plataforma sencilla, para su diseño se decidió poder contar con piezas que pudieran ser fabricadas en pequeña escala, aprovechando las herramientas comunes en el movimiento "*makers*" como son las impresoras 3d y máquinas de corte láser. Para el desarrollo de las piezas 3d se utilizó software libre (FreeCad) y una impresora de construcción casera modelo PRUSA I3 que permitió ir probando distintas alternativas de piezas que podían usarse en la plataforma V-plotter. Para ahorrar costes y tiempo de impresión, se decidió que las piezas más grandes podían ser fabricadas usando madera MDF (*medium density fiberboard*) y cortadas con máquinas de corte láser.

API (*Application programming interface*)

Para el cálculo de posición de la plataforma se toma el teorema del COSENO como herramienta matemática para poder obtener los pares ordenados que representan la posición X,Y relativa al punto (0,0) del cabezal. El esquema de trabajo de la API es el siguiente: recibe un vector de movimiento y un ángulo (en el esquema de trabajo LOGO), convierte esos vectores en pares ordenados x-y, para luego mediante un algoritmo de bresenham calcular el trazado de la línea de dibujo que tiene que seguir el cabezal. Luego decodifica la cantidad de pasos que debe dar cada motor que mueve las poleas y envía esa información en formato byte a la placa ICARO, que tiene conectada los dos motores mediante un integrado UNL2803 controlado por el PORTB del microcontrolador 18F4550. Como cada motor tiene 4 bobinas que se prenden secuencialmente para hacer un "paso", por tanto, para mover los 2 motores se usa un byte entero (8 bits) donde cada motor es controlado por 1 nibble (4 bits) del PORTB. De esta forma cuando la placa pone en estado alto los pines del PORTB, alimenta las correspondientes bobinas de los motores PAP 28BYJ-48 haciendo que este mueva su eje una posición.

Por su naturaleza, los V-plotters tienen un área de dibujo delimitada por las tensiones ejercidas por el peso del cabezal sobre los hilos de donde cuelga, además del diámetro de las poleas que recogen o sueltan cada hilo. A medida que el dibujo se va alejando del centro de la plataforma se puede apreciar una cierta deformación de las líneas rectas creando un

curva de forma ascendente, sin embargo a los efectos pedagógicos de generar dibujos usando lenguaje PYTHON, resultaron satisfactorios, por su bajo costo de fabricación y por cubrir correctamente las expectativas planteadas para la propuesta de enseñanza.

Consideraciones finales

El uso de dispositivos electromecánicos, como plataformas robots (Builes et al., 2011), se presentan como una herramienta muy poderosa para poder enseñar conceptos de programación (uso de repeticiones, variables, estructuras condicionales entre otras) generando situaciones genuinas de programación para los estudiantes. La capacidad de "abrir la caja negra" permite que los estudiantes ahonden en las tecnologías de base (Montfort et al., 2012) que permiten a su vez comprender cabalmente cómo son las interacciones entre hardware y software.

El desarrollo de esta plataforma V-PLOTTER, permitió al grupo de investigación dar cuenta de la necesidad de contar con recursos educativos abiertos (REA/OER) específicos para la didáctica de la programación (Bağcı, Kamaşak, & Ince, 2017), así como la posibilidad de retomar una currícula construccionista (Papert, 1993) trabajando con un lenguaje moderno como PYTHON (González Duque, 2016).

Bibliografía

Bağcı B.B., Kamaşak M. & Ince G. (2018). The Effect of the Programming Interfaces of Robots in Teaching Computer Languages. En Lepuschitz W., Merdan M., Koppensteiner G., Balogh R., Obdržálek D. (Eds) *Robotics in Education. RiE 2017. Advances in Intelligent Systems and Computing, vol 630*. Springer, Cham, pp. 88–99.

Busaniche, B., Chaparro, E., Heinz, F., Ribeiro, S.,Rodríguez Cervantes, S. Fiorito, S. & Westermann, W. (2007). *Monopolios artificiales sobre bienes intangibles*. Córdoba, Ediciones Fundación vía libre.

González Duque, R. (2016). *Python para todos*. Recuperado el 3 de marzo de 2019 de http://www.utic.edu.py/citil/images/Manuales/Python_para_todos.pdf

González, I., González, J., & Gómez-Arribas, F. (2003). *Hardware libre: clasificación y desarrollo de hardware reconfigurable en entornos GNU/Linux*. Recuperado de http://ftp1.nluug.nl/ftp/pub/ftp/os/Linux/doc/LuCaS/Presentaciones/200309hispalinux/8/8.pdf

González, Z., & HernándezII, G. M. (s. f.). Recursos educativos abiertos. Recuperado de http://www.medigraphic.com/pdfs/educacion/cem-2013/cem133p.pdf

Haché, A. (2014). *Soberanía tecnológica*. Dossier sobre Soberanía Tecnológica, pp. 9–18.Recuperado de https://www.plateforme-echange.org/IMG/pdf/dossier-st-cast-2014-06-30.pdf.

Jiménez Builes, J. A., Ramírez Patiño, J. F., & González España, J. J. (2011). Collaborative robotics modular system used in education. *Revista Facultad de Ingeniería Universidad de Antioquia*, (58), 163-172

Montfort, N., Baudoin, P., Bell, J., Bogost, I., Douglass, J., Marino, M. C. & Vawter, N. (2012). *10 PRINT CHR$(205.5+RND(1));? GOTO 10*. Cambridge (MA), The MIT Press. Leiva Olivencia, J. & Moreno Martínez, N. (2015). Recursos y estrategias educativas basadas en el uso de hardware de bajo coste y software libre: una perspectiva pedagógica intercultural. *Revista científica electrónica de Educación y Comunicación en la Sociedad del Conocimiento*, 15(1), pp. 30-50.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York, Basic Books, Inc., Publishers.

Papert, S. (1993). *The children's Machine: Rethinking School in the Age of the Computer*. New York, BasicBooks, Inc., Publishers.

Reyes Cortés, F. (2011). *Robótica: control de Robots manipuladores*. México, Alfaomega.

Stallman, R. M. (2007). *Software libre para una sociedad libre.* Madrid, Traficantes de Sueños.

Zabala, G. (2007). *Robótica*. Buenos Aires, Gradi.

Acercando la programación a la escuela secundaria con RITA

Isabel Miyuki Kimura, Vanessa Aybar Rosales, Claudia Queiruga[1]

Resumen[2]: Desde hace unos años el pensamiento computacional se propone como un enfoque de la enseñanza de la Informática en distintos niveles de la educación obligatoria, dejando atrás el enfoque de "informática de usuario". Nuestro país no es ajeno a ésto y múltiples políticas y programas federales dan cuenta de su importancia estratégicas entre otras cosas vinculada al desarrollo económico. Varios de los conceptos propios del pensamiento computacional se ponen en juego al programar, y esto viene aparejado con la necesidad de construir herramientas cercanas a la cultura adolescente.

La propuesta de RITA (Robot Inventor to Teach Algorithms) propone introducir el pensamiento computacional a través de la programación con un enfoque lúdico dirigido a adolescentes, con especial atención en el aprendizaje significativo a través de la resolución de problemas conocidos por los jóvenes y de trabajo en equipos.

RITA es una herramienta libre, de código fuente abierto, desarrollada en el LINTI-UNLP que propone a los estudiantes programar estrategias de robots virtuales que sean capaces de competir en un campo de batalla. El desafío consiste en programar estrategias de robots ganadores incorporando cierta "inteligencia" a los mismos. RITA es una herramienta que utiliza el enfoque de programación visual basada en bloques.

Las estrategias de los robots se programan con bloques organizados en múltiples categorías que le permiten realizar acciones en el campo de batalla, entre ellas desplazarse, reaccionar y responder a eventos que ocurren en su entorno, así como también realizar operaciones aritméticas, trigonométricas, lógicas, etc. Se aplican conceptos del campo de la Informática, específicamente de la programación: alternativas condicionales, repetición, uso de variables, abstracción y modularización a través de procedimientos, nociones de concurrencia y de programación orientada a eventos; y de matemática, nociones de ángulos y operaciones trigonométricas para la ubicación y desplazamiento en el plano de coordenadas cartesianas.

RITA pone a disposición la traducción de los bloques en lenguaje Java, ofreciendo a los docentes la oportunidad de introducir nociones de lenguajes de programación textuales.

RITA es utilizada y evaluada en forma continua en las escuelas que participan del Proyecto de Extensión del la UNLP "Extensión en Vínculo con Escuelas Secundarias". La propuesta de enseñanza con RITA contempla la creación y construcción de robots virtuales portadores de estrategias, que competirán con otros propiciando una situación de aprendizaje lúdico. RITA está disponible para su descarga desde https://github.com/vaybar/RITA

---

1 Laboratorio de Investigación en Nuevas Tecnologías Informáticas (LINTI), Facultad de Informática, Universidad Nacional de La Plata (UNLP).

2 Sin desconocer la pauta sexista del idioma español ni las novedosas prácticas de lenguaje inclusivo, a los fines de facilitar la lectura, en este trabajo se usará el genérico masculino al referirnos a las categorías de personas.

Introducción

En Argentina se han incorporado espacios para la enseñanza de robótica y programación que forman parte de la currícula de la educación obligatoria, ambas temáticas permiten trabajar en la escuela en la formación del pensamiento computacional (Queiruga C., 2017). La provincia de Buenos Aires incorporó en el año 2018, en el diseño curricular de educación primaria (DGCyE/Diseño curricular para la educación primaria de la provincia de Buenos Aires, 2018), las ideas del pensamiento computacional y la intención de incorporarlas transversalmente en las diferentes áreas curriculares (Queiruga C., 2018). La provincia de Neuquén es otro ejemplo de la adopción de este nuevo paradigma.

Actualmente existen múltiples herramientas disponibles de acceso online que favorecen el acercamiento a la programación de niños y jóvenes, y que están siendo adoptadas en instituciones escolares y en espacios de aprendizaje no formal, ejemplo de ellas son Scratch[3], Juegos de Blockly[4], Pilas Bloques[5], CodeCombat[6], además algunas de ellas ofrecen la opción de descarga para ser usadas de manera offline. Una características en común de varias de estas herramientas es el uso de la programación visual basada en bloques, al estilo de bloques de LEGO que se encastran en base a sus formas, permitiendo formular programas y evitando enfrentarse a los típicos problemas de un lenguaje de programación tradicional, textual, donde la primera barrera son los problemas de sintaxis.

Estas herramientas buscan, a partir de una consigna planteada como un problema o desafío, que el estudiante desarrolle una solución automatizada mediante un programa escrito en lenguaje de bloques visuales, usando conceptos de programación como secuencias, estructuras de control condicionales e iterativas, operadores lógicos, manejo de eventos, concurrencia, entre otros y desarrollando prácticas relacionadas a la abstracción, modularización y evaluación (Brennan y Resnick, 2012).

También existen otras herramientas que dan más libertad en las construcciones que el estudiante puede realizar, como MIT AppInventor[7], sin embargo, también es mayor el grado de complejidad en el uso, debido a la mayor cantidad de opciones disponibles y los conceptos que se pretende sean aprendidos por los estudiantes.

RITA busca mantener la simplicidad en la construcción de soluciones, al estilo de Pilas Bloques, haciendo énfasis en el trabajo y aprendizaje colaborativo en el aula, a partir del intercambio de experiencias en la puesta en común de las distintas soluciones. La propuesta de desafíos se enfoca en la construcción de estrategias de robots virtuales que compiten en un campo de batalla con otros robots. Estas estrategias o programas aunque pueden elaborarse de manera individual o grupal, requieren de una puesta en común, en el campo

3 https://scratch.mit.edu

4 https://blockly-games.appspot.com

5 http://pilasbloques.program.ar

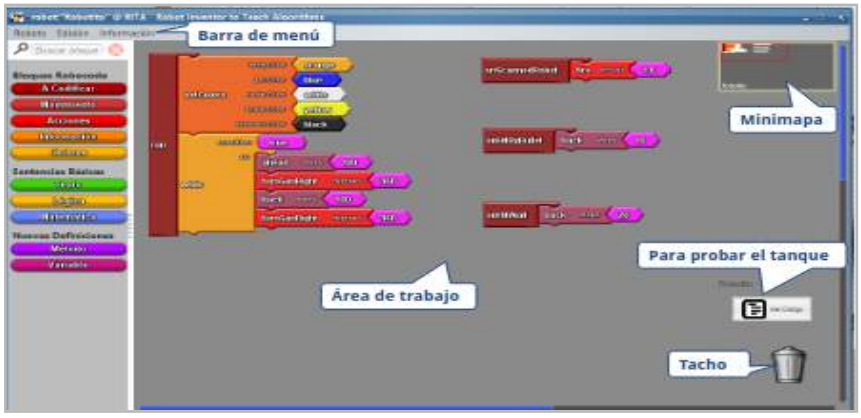
6 https://codecombat.com/

7 MIT AppInventor https://appinventor.mit.edu/

de batalla, de todas las estrategias desarrolladas, promoviendo el trabajo en equipos, el aprender “del otro” y la integración de los estudiantes. Esta puesta en común es una característica clave de RITA, que resulta altamente motivadora entre los estudiantes, promoviendo habilidades vinculadas a la comunicación, a la creatividad, al trabajo e interacción con pares, siendo éstas algunas perspectivas del pensamiento computacional, es decir, los puntos de vista que los estudiantes construyen sobre el mundo que los rodea (Brennan y Resnick, 2012). No es una novedad el uso de entornos competitivos relacionados a la programación, ya desde los años 80 existen proyectos que convocaban a participar en torneos de programación, ejemplo de ello es el juego de programación CoreWars[8].

En cuanto a la infraestructura de las escuelas de la provincia de Buenos Aires, si bien en estos últimos años se acercó nuevo equipamiento e incluso en algunos casos se crearon laboratorios con equipos, la conectividad a internet no está garantizada en todas ellas. Es por ello que RITA es una herramienta que puede ser descargada e instalada para no depender del condicionante de la conexión a internet (Aybar Rosales Vanessa, 2015).

RITA – Características



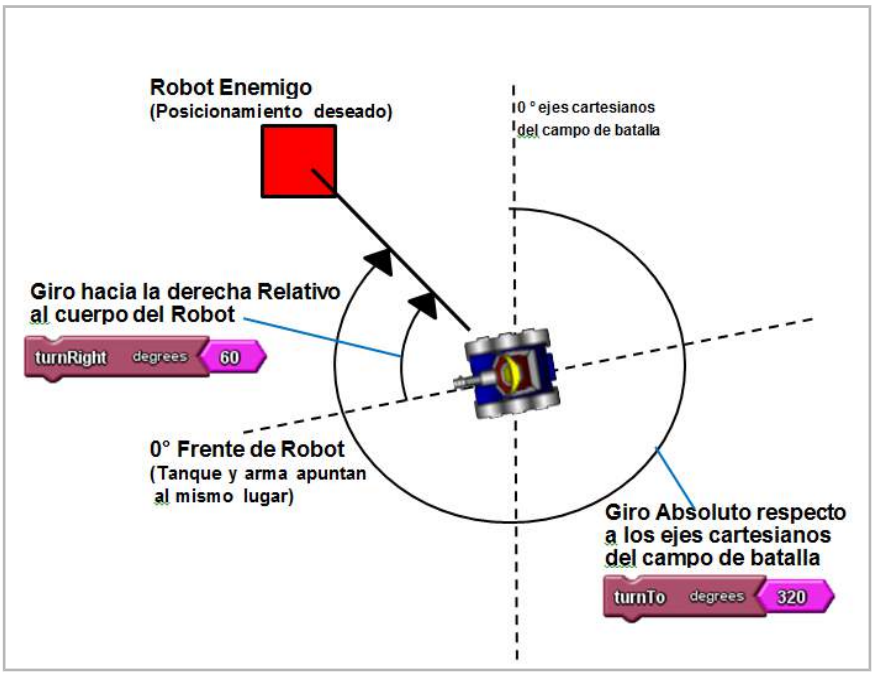
Área de trabajo en RITA.

RITA (Robot Inventor to Teach Algorithm) es una herramienta didáctica, cercana a la cultura adolescente que promueve la enseñanza de la programación mediante un abordaje lúdico.

Los bloques de RITA representan acciones que el robot realiza en un campo de batalla. Varias de estas acciones permiten a los robots desplazarse y reaccionar en respuesta a eventos que ocurren en su entorno. Asimismo, RITA cuenta con un amplio conjunto de bloques para realizar operaciones aritméticas, trigonométricas, lógicas, manipular variables, etc. El desafío propuesto por RITA es una competencia que alienta a los estudiantes a pensar en estrategias superadoras durante el proceso de programación de la estrategia de su robot, lo que mejora la calidad del aprendizaje. A su vez, en el proceso de programación de las estrategias se ponen en práctica conceptos de matemática como trigonometría,

8 Core War: http://www.corewar.info/tournament.htm

manejo de coordenadas cartesianas que los estudiantes aprenden en la escuela, articulando de esta manera contenidos disciplinares propios de la escuela secundaria.



Bloques que permiten el movimiento de un robot hacia una dirección y en un ángulo determinado.

RITA es una aplicación JAVA que extiende, integra y adapta las funcionalidades de dos frameworks también de código fuente abierto, Openblocks[9] y Robocode[10]. Openblocks brinda el soporte para la programación usando bloques gráficos que recrean el uso de piezas de LEGO.

En RITA, el framework Openblocks fue extendido para brindar soporte a las clases JAVA provistas por el framework Robocode, así como también proveer las estructuras de control, funciones y operadores de JAVA. De la misma manera que con las piezas de un LEGO, los conectores de OpenBlocks indican cómo pueden unirse los bloques. Los estudiantes comienzan programando con RITA simplemente encastrando bloques y obteniendo secuencias de bloques. De este modo el usuario de RITA puede programar la estrategia de un robot usando bloques gráficos sin preocuparse por la sintaxis y semántica del lenguaje JAVA. Automáticamente y en forma transparente para el usuario de RITA, la programación en bloques se traduce en código JAVA. Aquellos estudiantes que muestren curiosidad por visualizar un lenguaje de programación real, tienen a disposición en la interfaz gráfica de RITA la traducción de la composición de bloques en código fuente JAVA, el cual por el momento no es editable. Cabe destacar que los bloques en RITA se encuentran etiquetados en inglés y mantiene asociado un comentario explicativo en castellano. Esta decisión fue adoptada con el fin de familiarizar al estudiante con las palabras claves de los lenguajes de programación, sin embargo las experiencias nos han demostrado que suma complejidad y en las futuras versiones o actualizaciones se prevé contar con bloques en castellano.

9 *OpenBlocks: http://education.mit.edu/openblocks*
10 *Robocode: http://robocode.sourceforge.net/*

Posteriormente se trabajó en dos versiones de lo que llamamos RITA EN RED, ambas abordaron desde distintas soluciones la simplificación de la comunicación en red de RITA para el intercambio de las estrategias de los robots realizados por los estudiantes y facilitar la puesta en común de la competencia. La primera versión culminó en un trabajo de tesis con la realización de pruebas de campo y la segunda versión alcanzó la etapa de testeo[11].

Debido al avance de la tecnología, en la actualidad se está trabajando en la formulación de una nueva propuesta desarrollada con frameworks actualizados en lo que respecta a la herramienta que permite la construcción en bloques, y manteniendo el uso de Robocode. La continuidad del proyecto está prevista como trabajo de investigación y/o tesina de grado de estudiantes de la Facultad de Informática de la UNLP.

SECUENCIAS DIDÁCTICAS CON RITA

RITA se complementa con material dirigido a docentes y estudiantes. Desde el LINTI se elaboró una presentación donde se explica de modo incremental el uso de la herramienta y los bloques. Durante esta presentación se propone a los estudiantes la ejecución de algunos desafíos con el objeto de hacer una puesta en común al final de cada uno de ellos. Este material se encuentra disponible bajo licencia Creative Commons[12].

Adicionalmente se realizó una secuencia didáctica[13] que resulta en una guía para el docente donde paso a paso puede analizar las posibles alternativas y respuestas de los estudiantes a ejercicios propuestos. Del mismo modo que en la presentación se pretende que los conceptos se aprendan de un modo incremental. Esta secuencia didáctica cuenta con fichas para ser resueltas por los estudiantes y abarca los siguientes temas:

- uso de bloques en RITA
- introducción a los conceptos de secuencia y repetición de instrucciones
- conceptualización de reacción ante eventos
- uso de bloques de información para la toma de decisiones del robot
- creación de estrategias defensivas vs. ofensivas

Todo el material elaborado, tanto para éste como para otros proyectos relacionados a la enseñanza de programación están siendo organizados para ofrecerlos a través de un portal web. Este portal actualmente se encuentra en etapa de testeo.

---

11 *RITA en red está disponible para su descarga desde https://github.com/vaybar/RITA/tree/red*
12https://archivos.linti.unlp.edu.ar/index.php/s/jeJ5p0iWpt14yAN
https://archivos.linti.unlp.edu.ar/index.php/s/Wjzj0juMRslvFuK
13 https://archivos.linti.unlp.edu.ar/index.php/s/5ftRbgZpIU7kbPY

Experiencias con RITA

Desde el año 2012, el LINTI a través de proyectos de extensión e investigación, articula actividades con escuelas secundarias de la Provincia de Buenos Aires. Este vínculo facilitó la realización de actividades con RITA en escuelas, en general convocando a estudiantes del primer ciclo del secundario y realizando jornadas en el aula de clases, en donde los estudiantes aprenden a usar la herramienta, interactúan constantemente preguntando y analizando opciones en la preparación de la estrategia de su robot y la ponen a prueba.
En este mismo contexto, y a través del proyecto "Extensión en vínculo con Escuelas Secundarias" se continúa con la línea de trabajo de acercar la informática a la escuela, siendo el eje de programación uno de los más trabajados y se invita a los estudiantes de escuelas secundarias a conocer la Facultad de Informática y participar de una capacitación donde se trabaja usando RITA, en al menos 2 encuentros. En este caso, los alumnos convocados generalmente corresponden al segundo ciclo de secundaria, quienes podrían verse más interesados en conocer el ambiente de facultad.

Desde las primeras experiencias con RITA se relevaron a través de encuestas y al diálogo con los estudiantes, distintos aspectos a mejorar, como por ejemplo, la incorporación de un robot al que llamamos "Mambo" que permita acompañar al robot del estudiante, mientras resuelve una consigna determinada, o la necesidad de dar libertad al estudiante de posicionar a su oponente para probar una reacción particular de la estrategia. Además estas encuestas indicaron que en general el estudiante se sentía cómodo en el uso de la herramienta.

Durante las experiencias con estudiantes se pudo observar la integración del docente acompañando el trabajo de los estudiantes, por otro lado respecto a la cuestión de género en el uso de la herramienta y la temática involucrada, resultó indiferente, entendemos que el abordaje lúdico propicia la motivación de todos.

Conclusión

Hoy en día la incorporación de la informática en los diseños curriculares de la educación obligatoria continúa siendo un tema de debate, algunas provincias de nuestro país han logrado avanzar y realizado acuerdos en relación a un enfoque alineado al pensamiento computacional. La disposición de materiales y herramientas que permitan acercar la programación a la escuela ha evolucionado notablemente en esta última década, siendo por el momento la programación visual basada en bloques la más aceptada. Desde el LINTI se presenta RITA como propuesta de herramienta para la enseñanza de la programación y formación del pensamiento computacional, junto con material, destinado para el docente, compuesto por presentaciones para facilitar el dictado y fichas para los estudiantes, para su utilización y adecuación.

Bibliografía

Aybar Rosales Vanessa, Queiruga Claudia, Kimura Isabel, Brown Barnetche Matías y Gómez Soledad (2015). “Enseñando a programar con RITA en escuelas secundarias”. En *XXI Congreso Argentino de Ciencias de la Computación* (CACIC 2015), Junín, Argentina, Octubre 5-9, 2015. ISBN 978-987-3724-37-4, pp. 1201-1211.

Aybar Rosales Vanessa, Queiruga Claudia, Banchoff Tzancoff Claudia, Kimura Isabel Miyuki y Brown Bartneche Matías (2017). “Programming Competitions in High School Classrooms: RITA en RED”. En 2017 *XLIII Latin American.* Córdoba, 4 al 8 de septiembre de 2017. Editorial: IEEE. ISBN: 978-1-5386-3057-0. Indexada: DBLP, IEEE Xplore.

Brennan, K., y Resnick, M. (2012). “New Frameworks for Studying and Assessing the Development of Computational Thinking” (Nuevos Marcos de referencia para estudiar y evaluar el desarrollo del Pensamiento Computacional). In: *Proceedings of the 2012 annual meeting of the American Educational*. Research Association, Vancouver, Canada.

Queiruga C., Banchoff Tzancoff C., Venosa P., Martin S., Aybar Rosales V., Gomez S., Kimura I. (2018). “Escuelas TIC: las tecnologías digitales en las aulas”. En *WICC 2018* (Workshop de Investigadores en Ciencias de la Computación). UNNE, Corrientes, Argentina. Abril de 2018. Libro de Actas XX Workshop de Investigadores en Ciencias de la Computación. . pp 475-489. ISBN 978-987-3619-27-4.

Queiruga C., Banchoff Tzancoff C., Martin S., Aybar Rosales V., López F., Kimura I. y Gómez S. (2017). “PROGRAMAR en la Escuela: Nuevos Desafíos en las Aulas”. XIX En *Workshop de Investigadores en Ciencias de la Computación* (WICC 2017), Ciudad de Buenos Aires, 27 y 28 de Abril. ISBN 978-987-42-5143-5, pp 732-736. Editorial: Red de Universidades con Carreras en Informática (RedUNCI).

Wollok: Un entorno de aprendizaje de Programación Orientada a Objetos

Lucas Spigariol[1]

Introducción

Wollok es un software educativo para aprender a programar basado en el paradigma de la Programación Orientada a Objetos, desarrollado por docentes e investigadores de Universidades Públicas de Argentina, en el marco de la Fundación Uqbar. Desde una mirada tecnológica, se trata de un *entorno de desarrollo de software* que incluye un lenguaje de programación propiamente dicho, con un alcance conceptual similar a los lenguajes de alto nivel de uso profesional en la actualidad. Desde una mirada educativa, es un *entorno de aprendizaje* que propone una manera particular de llevar adelante un proceso educativo en programación [PFTD17], basándose en una herramienta concreta que permite articular fecundamente teoría y práctica.

La decisión medular que llevó a su desarrollo y utilización parte de priorizar las opciones pedagógicas por sobre las limitaciones tecnológicas. En otras palabras, en vez de diseñar itinerarios formativos que se adapten a las características que tienen las herramientas de desarrollo de software de uso industrial, se optó por desarrollar una herramienta tecnológica que se adapte a una concepción de qué, cómo y por qué enseñar a programar. Las principales características que presenta apuntan a abarcar un abanico de situaciones que van desde estudiantes sin experiencia previa a quienes se quiere introducir en el pensamiento computacional (interfaz gráfica interactiva, diagramas de objetos, abstracciones simples, estructuras de control básicas, objetos autodefinidos, reportes de errores en español, etc.) hasta estudiantes que tienen como meta su inserción laboral en el ambiente de desarrollo de software (tests unitarios, integración con *Git*, manejo de clases, herencia y mixins, sistema de inferencia de tipos, etc.)

Las forma recomendada de usar Wollok es como aplicación de escritorio, aunque existe una alternativa de ejecutarlo por línea de comando de reciente creación (Wollok CLI) y también es posible usarlo desde la plataforma educativa on line *Mumuki*, donde hay guías interactivas de ejercicios, problemas y desafíos que ayudan a sostener el proceso de aprendizaje.

1 Universidad Tecnológica Nacional, Universidad Nacional General San Martín, Fundación Uqbar
lspigariol@gmail.com

Origen y fundamentación

A pesar de ser la forma de construir software más utilizada en el ambiente profesional, la Programación Orientada a Objetos no suele ocupar un lugar significativo en las propuestas pedagógicas sobre programación. Esta decisión puede tener diferentes motivaciones, entre ellas la percepción de que es difícil de enseñar [BC04, Jen02, Uys], porque requiere del estudiante la capacidad de manejar muchos conceptos incluso para hacer soluciones sencillas, o que suele ser necesario aprehender un volumen considerable de teoría antes de poder hacer una práctica concreta [KQPR03, SGURGH15], lo cual lleva a considerarlo como un contenido lejano a la educación básica y media, e incluso en los trayectos de educación superior se la suele presentar como un contenido más avanzado.

La convicción que condujo a crear *Wollok* es que la dificultad no está en el paradigma en sí, sino en la elección de recorridos didácticos que no permiten una incorporación gradual de sus nociones centrales a lo largo del curso y en cambio obligan a que un estudiante sea capaz de manejar un amplio abanico de conceptos teóricos antes de poder producir su primer programa [CO16]. Estos recorridos, sin embargo, están frecuentemente limitados por la utilización en el ámbito del aprendizaje de lenguajes de programación y otras herramientas pensadas para profesionales, que naturalmente priorizan maximizar el rendimiento de un experto por sobre facilitar el acceso a un principiante [ACS02, GF03].

Frente a esta dificultad, la reflexión pedagógica fue que la secuencia didáctica debía articular teoría y práctica desde un primer momento y que no debía restringirse debido a las herramientas tecnológicas existentes, sino al revés: que el lenguaje de programación sea el que vaya acompañando y sosteniendo el itinerario formativo.

Un conjunto de asignaturas de programación en carreras universitarias de informática, donde confluyen la percepción de la dificultades mencionadas en los procesos de aprendizaje de los estudiantes con la capacidad profesional de los docentes de poder crear los propias herramientas tecnológicas, constituyó un espacio fecundo para el surgimiento de *Wollok*, fue un ambiente contenido para probar su funcionamiento, es el origen de una lluvia de ideas para mejorarlo permanentemente y sigue siendo un semillero de vocaciones de investigación que creen que tiene sentido apostar a un desarrollo tecnológico autónomo y de calidad desde la Universidad y el Estado.

En su origen, el perfil para el que se pensó la herramienta es el de un estudiante universitario sin experiencia previa en programación orientada a objetos. La realidad de los planes de estudios de las carreras universitarias en las que se comenzó a utilizar *Wollok* condiciona que quienes llegan a las asignaturas donde se aprende a programar en el paradigma de objetos tienen como correlativas previas materias donde se enseñan nociones básicas de programación, con mayor o menor intensidad y utilizando diferentes lenguajes según la institución, pero en todos los casos en términos de programación procedural. De esta manera, conceptos habitualmente considerados iniciales como por ejemplo estructuras de control condicionales, manejo de variables y constantes o paso de parámetros, que sin

duda se podrían explicar utilizando *Wollok* si el contexto fuese otro, no se lo hace porque no es necesario y los docentes se limitan a especificar la sintaxis con la cual se lo expresa. Incluso más recientemente, con la incorporación de *Wollok* en escuelas secundarias técnicas, se da una situación similar ya que si bien son estudiantes de menor edad, también pasan previamente por materias de otras formas de programar antes de abordar el paradigma de objetos.

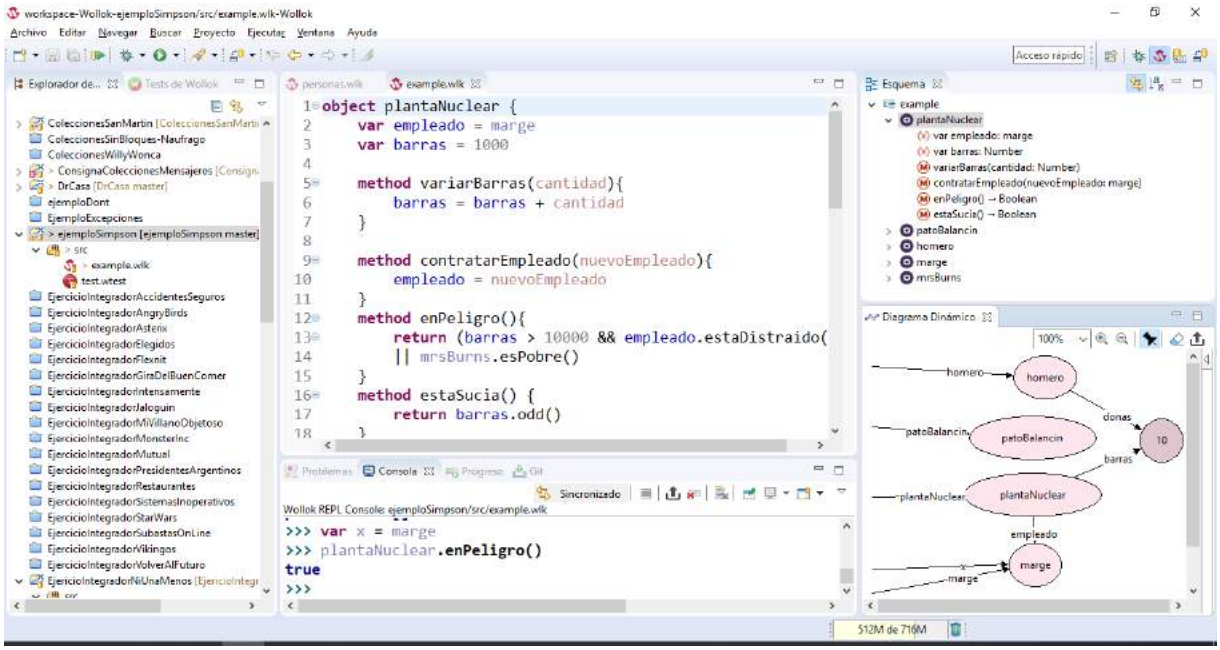
No es el objetivo de este trabajo establecer un discusión acerca de si la mejor forma de introducir a los estudiantes en el mundo de la programación es en términos de la programación procedural o de objetos -o por qué no utilizando conceptos de otros paradigmas como la programación funcional o lógica-, pero sí se busca que, para quien quiera experimentar que la primera aproximación a la programación sean en con objetos, conozca una herramienta concreta que le permita hacerlo, evitando las complejidades que tienen las herramientas profesionales de desarrollo de software en este paradigma, que están diseñadas para aumentar la eficiencia de quien ya sabe y no para facilitar el aprendizaje como se propone *Wollok*.

Wollok se viene utilizando desde hace cinco años en materias iniciales de programación en diferentes universidades (actualmente en UTN, UNGSM, UNQ, UNDAv, UNaHur) y también en algunas escuelas medias de orientación técnica.

Como antecedente directo, se desarrolló y utilizó durante unos años también una herramienta de software llamada Ozono, cuyo objetivo y fundamentos pedagógicos eran similares, pero que en vez de ser un lenguaje en sí mismo se ejecutaba dentro del ambiente de desarrollo de Smalltalk y se preveía su uso durante parte del curso para luego dejarlo y continuar con Smalltalk propiamente dicho. Su uso fue satisfactorio [Spi15], tuvo varias denominaciones a medida que cambiaba de versión (Object Browser, Loop), pero se discontinuó su uso ante la existencia de *Wollok.*

Descripción de la herramienta

Como primera aproximación, *Wollok* es un lenguaje de programación en el sentido típico del término: Con él se pueden escribir programas, se utiliza una determinada sintaxis, su código se ejecuta, se modelan soluciones y se obtienen resultados. Técnicamente es un intérprete y corre sobre diferentes sistemas operativos; desde una mirada docente, abarca los conceptos centrales del paradigma de objetos (objetos, mensajes, atributos, clases, colecciones, polimorfismo, herencia, redefinición) como también otros que se pueden considerar como más avanzados o que rondan los límites del paradigma (mixins, interfaces) y a la vez incluye elementos transversales a los diferentes paradigmas de programación (tests unitarios, manejo de excepciones, interfaz gráfica).



Vista general de Wollok, en la que se ve una porción de código con objetos y métodos, la consola REPL y el diagrama dinámico.

Hilando más fino, *Wollok*, más que un *lenguaje de programación*, es un *entorno de aprendizaje y desarrollo de software*. Existe una concepción en ciertos ambientes académicos que asocian casi unívocamente la tarea de programar al uso de un lenguaje de programación; en otras palabras, que ve al programa escrito como un producto y a su proceso de desarrollo como una tarea de escritura. En este enfoque, el ambiente de desarrollo es un simple editor de texto -que puede ser fácilmente sustituible por el pizarrón en una clase y por una hoja de papel en un examen- y que en el mejor de los casos viene complementado con el compilador adecuado que a modo de caja negra transforma ese código en un programa ejecutable. En caso que el código sea correcto, se ejecuta el programa y se observan los resultados, en caso que haya errores, el compilador advierte sobre su ubicación y naturaleza para que el programador reescriba los cambios necesarios y vuelva a compilar. En la dinámica moderna de desarrollo de software, se destacan los entornos integrados de desarrollo (IDEs) que no sólo permiten escribir código, sino que incluyen diversas herramientas que facilitan la tarea de programar. En este sentido, *Wollok* viene integrado en el entorno Eclipse, con una configuración especialmente diseñada en la que se aprovechan algunas de sus características, se ocultan otras para evitar complejidad innecesaria y a la vez se incorporan las específicas del lenguaje.

Puntualmente, algunas de estas características que facilitan el aprendizaje, y que lo diferencian de otras herramientas, son:

- Detección de errores, advertencias y sugerencias a medida que se escribe el código, lo que brinda un feedback instantáneo.

- Consola REPL para consultas interactivas, sin necesidad de salir del entorno de desarrollo.
- Integración con herramienta para definir y ejecutar pruebas unitarias.
- Diagramas que se generan automáticamente: El estático, que se basa en el código del programa escrito y muestra la estructura de los objetos y clases, sus atributos y métodos, las relaciones de herencia, entre otras características. El dinámico, que se va modificando conforme se van ejecutando diferentes partes del programa desde la consola interactiva. Ambos diagramas son representaciones gráficas de conceptos fundamentales en el paradigma, que acompañan y refuerzan su comprensión.
- Interfaz gráfica para hacer videojuegos.
- Sistema de inferencia de tipos, que permite realizar sugerencias y detectar inconsistencias entre la definición y uso de los diferentes módulos del programa, sin necesidad de un tipado explícito.

Pero para dar un paso más acerca de lo que es *Wollok*, y el sentido que tiene contar con estas herramientas para llevar adelante un proceso de aprendizaje, cabe hacerse algunas preguntas sobre qué y cómo enseñar.

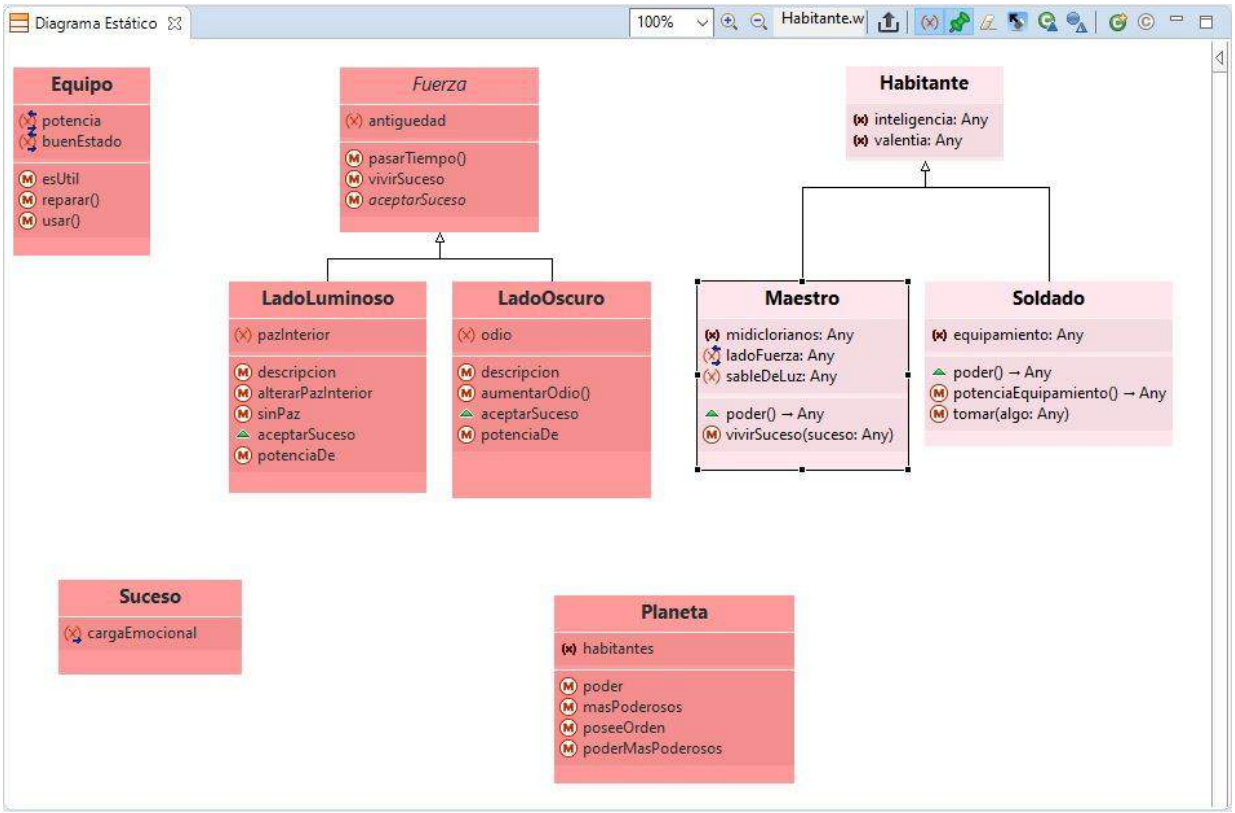
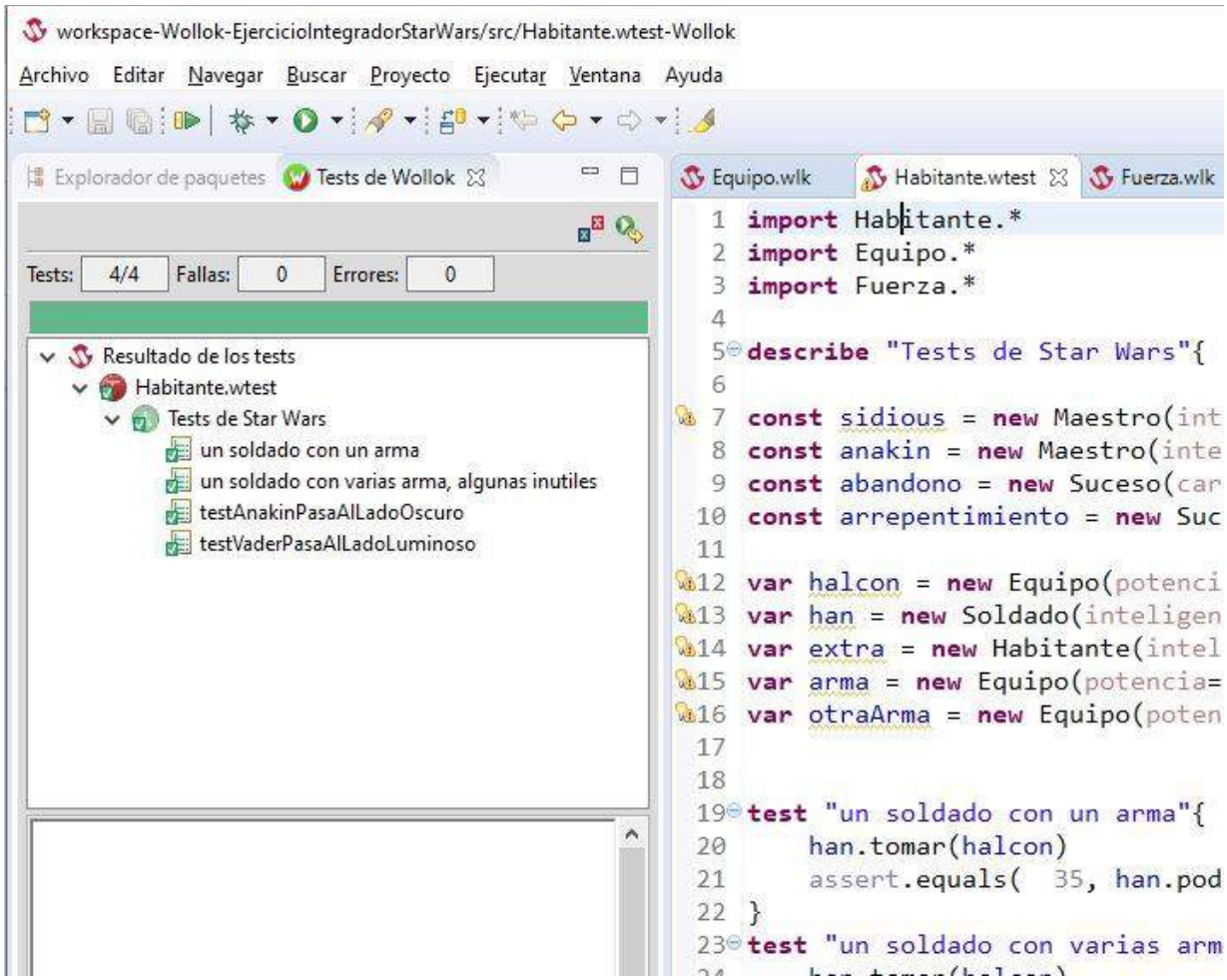


Diagrama estático de un ejercicio, donde se ven las clases con sus atributos y métodos y las relaciones de herencia.



Herramienta para pruebas unitarias. Resultado de la ejecución y porción de código de su definición.

¿Qué enseñar?

Dentro del amplio abanico de nociones y destrezas propias de la programación, tanto de las cuestiones típicas de la programación orientada a objetos como de otras formas de programar, se realiza una cierta selección y ponderación.

Se destaca una de las intuiciones originarias del paradigma de objetos de disminuir el *gap semántico* entre la realidad y el modelo, es decir, desde la perspectiva del estudiante, que las entidades que están en el programa le resulten lo más cercanas y representativas posible a las entidades del mundo real en que se basan. Por esto, se presenta al *objeto* como elemento central, que más que una entidad inerte o simple contenedora de información se lo interpreta como *sujeto*, con comportamiento e identidad, que interactúa con otros, que *conoce* mediante sus atributos, *pregunta* al enviar mensajes y *colabora* al ejecutar un método. A su vez, al objeto se lo rescata en su originalidad e individualidad, sobre la que eventualmente en un paso posterior se puede generalizar a partir de descubrir similitudes con otros, antes que pensarlo como un caso particular de una generalización abstracta de existencia previa y alcance predeterminado.

Pensar a un programa de computación como un conjunto de *objetos/sujetos* que interactúan, es una forma de modelar la realidad que, además de constituir una metáfora de la sociedad que dispara miles de preguntas y conecta con otras disciplinas, resulta relativamente sencilla de comprender y constituye un forma de pensamiento computacional con diferentes niveles de complejización.

En concreto, *Wollok* permite programar fluidamente con *objetos autodefinidos* sin necesidad de recurrir inicialmente al concepto de *clase,* como suele suceder en las propuestas pedagógicas existentes. Esto va de la mano de la intención de mostrar las virtudes del paradigma de objetos sin caer en las inercias procedurales que generalmente provienen de entender la *clase* como una simple estructura de datos. Además, que los objetos tengan un identificador conocido permite trabajar con ellos sin preocuparse aún por el manejo de referencias y el uso de variables, como sucede en lenguajes más tradicionales como Smalltalk donde no se puede interactuar con un objeto sin antes tener una referencia.

Otra de las premisas de *Wollok* es la de un tipado dinámico que permite un manejo de *polimorfismo* independientemente de la *herencia*, *interfaces*, *clases* ni ninguna definición de tipos de variables. La convicción acerca de la importancia del *polimorfismo* como forma de entender las interacciones entre objetos, tanto como metáfora de la no discriminación y la capacidad de ponerse en el lugar del otro, como de su centralidad a la hora de construir soluciones informáticas profesionales, lleva a que se permita su utilización con un mínimo de elementos conceptuales previos y se lo ponga inmediatamente en práctica. A su vez, la facilidad que representa para un estudiante no tener que definir previamente el *tipo* de todo elemento del código, se complementa con un poderoso s*istema de inferencia de tipos* que informa sobre la formas de uso correctas y advierte posibles errores.

Aún sin ser lo que sucede dada la utilización actual de *Wollok* en situaciones donde se busca otra cosa, características tales como la forma de definir y asignar en variables, la posibilidad de tener una lista de elementos mediante una sintaxis sencilla y poder mezclarlos sin estar condicionados por su tipo, la existencia de un *if* y operadores lógico y matemáticos, también habilita su uso para quien no quiera profundizar en conceptos más específicos del paradigma de objetos.

¿Cómo enseñar?

Hay varios tópicos en los que se podría dar una respuesta. Se mantienen criterios de simplicidad en la sintaxis y facilidades de usabilidad en el entorno de desarrollo, se establece una fuerte asociación entre conceptos y palabras reservadas, se cuida la expresividad y localización de los mensajes de error que se suponen frecuentes en un estudiante que da sus primeros pasos, entre otras. Pero en particular se destacan dos características, que a su vez habilitan a diferentes formas de uso: el ordenamiento de la secuencia didáctica y la utilización de herramientas gráficas.

Secuencia didáctica

Desde una perspectiva de itinerario, el recorrido propuesto para el cual *Wollok* brinda facilidades, se organiza en cuatro etapas con temas centrales de complejidad creciente (*objetos, colecciones, clases y herencia*) y en todas ellas hace énfasis en la importancia del *polimorfismo*.

Una facilidad, sobre todo para los primeros ejercicios en objetos con los que se pretende aprender a modelar *objetos* y *mensajes*, es la posibilidad que los objetos tengan valores iniciales para sus variables sin que se les envíe ningún mensaje previamente. Permite además que los primeros mensajes que se usen sean sin *parámetros* y que en un posterior ejemplo se los incorpore. También, se ve la secuenciación en el manejo de *colecciones*. En otros lenguajes, para tener una *colección vacía* a la que se le pueda ir agregando elementos se debe manejar el concepto de *clase*, instanciarla indicando su nombre y enviándole un *mensaje de clase* llamado *new*. Lo que permite *Wollok* es una inicialización de colecciones sin necesidad de *métodos*, detallando los valores necesarios junto con la definición de las variables, y una sintaxis sencilla que recuerda a la notación de listas de otros lenguajes y conjuntos matemáticos. De esta manera, si el docente así lo prefiere, puede trabajar tranquilamente con *colecciones* antes que explicar el concepto de *clase*.

En muchas situaciones como esta, el docente se encuentra en la tensión entre adelantar la presentación de otros conceptos aún para hacer cosas simples, o a requerir de los estudiantes un acto de fe: por más que no entienda qué es o qué significa una cierta expresión, "hay que escribirla así para que funcione y listo". Frente a esto, *Wollok* evita incluir en su formulación expresiones arbitrarias o innecesarias, mantiene una marcada economía sintáctica sin sacrificar expresividad y, sobre todo, sugiere que en su código no se anticipen respuestas a problemas que aún no hayan surgido. Con criterios similares, el *polimorfismo* no está condicionado por la *herencia*, la *instanciación* no requiere de un *constructor* definido previamente, los objetos se pueden autorreferenciar sin necesidad de usar *self* o similar, y hay variantes para expresar el manejo de *excepciones* que permite introducir el concepto en diferentes momentos de la secuencia didáctica, entre otras decisiones.

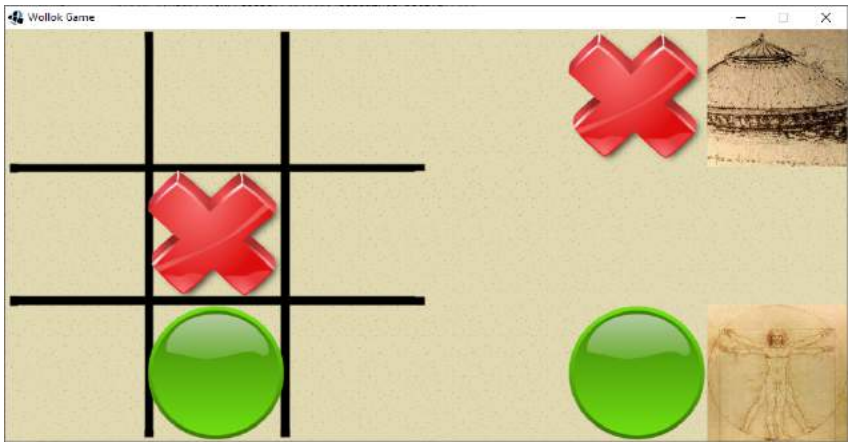
Elementos gráficos

Desde una perspectiva de proyectos, otra forma de utilización es mediante *Wollok Game*, una interfaz gráfica que viene integrada en la herramienta, que permite construir juegos y aplicaciones interactivas de menor o mayor complejidad y desde allí ir descubriendo y aplicando los diferentes conceptos del paradigma. Se basa en la asociación de los objetos presentes en el código con imágenes y posiciones en la pantalla de manera que la ejecución del programa sea visible a la vez que permite al estudiante interactuar con su creación como un usuario, vinculando los eventos con el código. Incluye facilidades propias de los

videojuegos tales como los desplazamientos y colisiones de elementos visuales, el manejo del tiempo, la interacción con el usuario final, la incorporación de música y sonido.

Otro aspecto visual importante, son los ya mencionados diagramas estáticos y dinámicos, que se generan automáticamente desde el código. El diagrama dinámico muestra el estado del sistema en todo momento reflejando las relaciones entre objetos y su evolución en el tiempo. Permite, por ejemplo, ver cómo las variables son referencias que apuntan a otros objetos, cómo al asignarle un valor la flecha del diagrama se mueve, cómo un objeto puede ser referenciado desde muchos lugares a la vez. Complementariamente, el diagrama estático muestra las clases y objetos definidos, permite que el estudiante visualice de un pantallazo general la organización de lo que está escrito en el código, comprenda mejor lo que está haciendo y pueda modificarlo con mayor agilidad. Se basa en el estándar de UML, refleja las relaciones definidas explícitamente, como las de herencia, y la potencialidad -es un feature en desarrollo- de integrarse con el sistema de tipos e inferir otras relaciones, como las de asociación o composición.

Ventana de un juego sencillo, hecho con Wollok Game por un grupo de estudiantes.



Una de las ventanas de un juego más complejo, hecho con Wollok Game por estudiantes, con diferentes niveles, personajes, herramientas y una lógica compleja.

Resultados

El feedback que se recibe de docentes y estudiantes es significativo y variado. En primer lugar están los reportes de *bugs,* que no sólo son útiles para resolver los problemas de implementación específicos a los que hacen referencia, sino que en cierto modo son un termómetro de la intensidad de su uso.

Más interesantes son las propuestas de mejoras y pedidos de nuevos features, como así también los comentarios sobre la aceptación o desacuerdo con las funcionalidades que se van agregando año tras año, que enriquecen la dinámica de desarrollo del proyecto y permiten ver la ampliación de docentes e instituciones que deciden utilizar o al menos probar *Wollok*.

Las evaluaciones y encuestas de los estudiantes sobre el desarrollo de los cursos en los que se enseña a programar con *Wollok* son generalmente muy buenas. Esto obviamente habla más de los respectivos docentes y su forma de llevar la materia que del lenguaje en sí, pero al menos muestra que su uso no es un motivo fuerte de desaprobación que arrastre la valoración de la asignatura. Cuando se trata de instrumentos elaborados en forma genérica por la institución educativa no es posible, pero en algunos casos en que se trata de una iniciativa del mismo docente y el formulario incluye alguna pregunta directa sobre *Wollok*, -o cuando se hace un balance de la cursada en forma oral- el resultado es ampliamente positivo. Las críticas que se encuentran, dejando de lado ciertos comentarios en referencia a bugs o situaciones puntuales, generalmente van por el lado de que se hubiese preferido un lenguaje comercial/industrial/profesional. La opinión de los mismos estudiantes como sujetos de su propio proceso de aprendizaje es valiosa como indicador de la utilidad de la herramienta pero no determinante por sí solo. Saliendo de los extremos de un eventual rechazo mayoritario o de una inimaginable actitud de admiración que no se constatan en la realidad, lo más valioso del feedback de los estudiantes es la posibilidad de hacer mejoras o ajustes dentro de la misma propuesta. Sin descartar que algunos con un poco más de experiencia lo puedan hacer, no se espera que el estudiante promedio tenga la suficiente distancia para analizar cómo hubiera sido el desarrollo de la materia utilizando otro lenguaje y sacar una conclusión al respecto. (Al menos al momento de concluir la materia, sería interesante su punto de vista años después, ya inmerso en el mundo profesional, habiendo tenido experiencias de otras herramientas de desarrollo).

Otro camino posible para validar científicamente el impacto de *Wollok* en el proceso de aprendizaje es entrar en el terreno cuantitativo de los análisis comparativos. Aún reconociendo las limitaciones de su implementación, lo discutible de sus eventuales resultados y ciertas discrepancias epistemológicas acerca de este tipo de metodologías en el ámbito educativo, hay algunos estudios en marcha de los cuales no se disponen resultados, pero que se espera puedan enriquecer el análisis en un futuro.

Teniendo en cuanta las consideraciones, posibilidades y limitaciones mencionadas, para el presente estudio, los indicadores que se consideran prioritarios a la hora de plasmar el

resultado de utilizar *Wollok* como herramienta que da soporte a la propuesta formativa son la convicción de parte de los docentes y el aumento de complejidad de las producciones de los estudiantes y el aumento de exigencia de las evaluaciones sin disminuir los niveles de aprobación. Ambos combinan un componente cualitativo importante sujeto a la intepretación con datos mensurables.

Convicción docente

De todos los docentes que habiendo utilizado antes otras herramientas pasaron a usar *Wollok*, ninguno volvió atrás o buscó otras alternativas superadoras, sino que aún reconociendo limitaciones apuestan a mejorarlo. Para dimensionar el escenario, la institución que más docentes -y estudiantes- tiene utilizándolo es la Facultad Regional Buenos Aires de la Universidad Tecnológica Nacional. Cada año, son alrededor de 15 comisiones las que se llevan adelante, cada una con su respectivo docente a cargo más un conjunto de ayudantes, que juntos conforman un equipo docente de más de un centenar de integrantes; sumando a todas las demás instituciones educativas, el volumen total aproximadamente se duplica. Más allá de los pormenores, como que la decisión de comenzar a usar *Wollok* no fue exenta de discusiones y objeciones iniciales, que en algunas instituciones fue tomada antes que en otras, que hay algunos docentes que trabajan en más de una institución a la vez, que en los equipos docentes hay una minoría que participa en el desarrollo mismo de *Wollok* y una mayoría que sigue el proceso a la distancia, con mayor o menor grado de cercanía, que los plantes de estudios vigentes no estipulan qué lenguaje utilizar o no, que casi la totalidad de los docentes trabaja a su vez en la industria del desarrollo de software y conoce de primera mano otros lenguajes de programación orientada a objetos, que existe la libertad de cátedra en todas las instituciones como para dar marcha atrás o tomar otros caminos, lo que sucedió es que todos los docentes involucrados lo continúa utilizando.

Complejidad de las producciones de los estudiantes

Se constata un aumento significativo de la complejidad de los exámenes y trabajos prácticos que realizan los estudiantes, lo cual va de la mano de la ampliación de los alcances temáticos de las planificaciones de las asignaturas, sin aumentar el tiempo neto de clase ni variar significativamente los porcentajes de aprobación.

Comenzando por esto último, es frecuente ver en las planificaciones actuales clase enteras destinadas a discusión sobre modelado en objetos y elementos de diseño, con temas tales como "Herencia vs Composición", "Clase abstracta e Interfaz", "Mutabilidad. Igualdad e identidad", "Template method", "Acoplamiento", "Introducción al Strategy", que antes no estaban presentes o eran mencionados lateralmente. Se agrega también como tema importante el testeo automatizado y el manejo de excepciones, con clases destinadas a "Testeo unitario automatizado avanzado", "Fixture y Clases de equivalencias " o "Lanzar y capturar excepciones".

Sobre los límites del paradigma, otros temas que aparecen al final del recorrido del curso, y con diferencias según el docente, el año o la institución, apuntan a hacer comparaciones con lenguajes industriales, establecer vínculos con otros paradigmas de programación, introducir a conceptos más novedosos dentro del mundo de objetos -por ejemplo, mixins, que si bien *Wollok* los implementa, habitualmente no forma parte del contenido curricular de las asignaturas- o profundizar algún concepto puntual como esquemas de tipado. Cabe señalar que este tipo de clases ya se realizaban en algunos casos antes de que se utilice *Wollok*, pero en mucho menor cantidad que los que sucede en la actualidad.

Esta profundización y ampliación de los alcances temáticos, que en buena medida es el correlato del menor tiempo requerido para que se comprendan las nociones básicas, se traduce en exámenes, ejercicios y trabajos prácticos de mayor complejidad. Siendo una variable de análisis de dificultosa cuantificación, el criterio elegido es la interpretación de los propios docentes, que en definitiva son quienes hacen un ejercicio subjetivo similar a la hora de definir la aprobación de cada uno de ellos. Un conteo de líneas de código en los repositorios donde se suben los trabajos o de la cantidad de clases y objetos que intervienen en la solución podría hacerse, y probablemente mostraría que el volumen aumentó, pero es más valioso analizar la complejidad del modelo implementado y los criterios de validación. Con mirada retrospectiva, lo que mayoritariamente ellos constatan es que los exámenes son realmente más difíciles, incluyen más temas o son más exigentes a la hora de establecer el mínimo de aprobación, prestando mayor atención al modelado, a las abstracciones formuladas, la expresividad del código o la economía del código. Una minoría no percibe mayores diferencias, pero en ninguno de los casos consultados se manifiesta una disminución. Otro aspecto que algunos señalan, sobre todo en los trabajos prácticos con consignas más abiertas de desarrollo, es la grata sorpresa de encontrarse con producciones de gran nivel.

En paralelo, y considerando en esta oportunidad lo datos estadísticas de porcentajes de aprobación -tomados por institución y por docente-, no han variado significativamente desde que se usa *Wollok* en relación a lo que sucedía antes. Los pequeños matices se distribuyen de manera irregular por lo que se atribuyen a otros factores y no al uso del lenguaje.

Conclusiones

Sin dejar de ser un *lenguaje de programación*, o un *entorno de aprendizaje y desarrollo de software Wollok* es una propuesta pedagógica para la enseñanza de la programación orientada a objetos. En otras palabras, es el *andamiaje* que sostiene y posibilita un proceso de construcción de conocimiento. Ciertamente, existen numerosos lenguajes y herramientas con las que se programa profesionalmente en objetos y todas ellas se pueden utilizar para enseñar. Pero ante las preguntas planteadas anteriormente de *qué* y *cómo* enseñar, sumando el *para qué* hacerlo, las posibilidades o facilidades que ofrece cada uno de los lenguajes existentes es diferente. Precisamente, lo que da sentido a *Wollok* es una respuesta pedagógica que es previa a la creación de la herramienta y constituye su inspiración.

Cuando se piensa la enseñanza del desarrollo de software como uno de los ejes centrales de una formación profesional, una disyuntiva que aparece, y que condiciona los objetivos y las herramientas para abordar la programación orientada a objetos en particular asumiendo que es una etapa dentro de un recorrido más amplio, es entre tener una propuesta de formación relativamente más reducida que apunte una salida laboral rápida o una propuesta más extensa que permita transitar con mayor detenimiento los temas y conceptos involucrados. Más allá que sea una discusión con mucha aristas y que en la actualidad conviven trayectos académicos que se orientan en uno y otro sentido, no deja de ser una discusión que suele aparecer en los equipos docente a la hora de definir una planificación y elegir las herramientas a utilizar, y muchas veces no tiene que ver directamente con que sea una licenciatura, ingeniería o tecnicatura, o que la institución que la lleva adelante sea una universidad, un instituto terciario o un secundario.

Si lo que se busca es que el estudiante tenga rápidamente las mínimas herramientas para buscar una posibilidad de inserción laboral, probablemente *Wollok* no sea una buena opción, difícilmente un seleccionador de personal valore ese ítem en un CV y el estudiante mismo sea consciente que aún habiendo aprendido conceptos de programación le falte hacer una cierta transición e incorporar otros elementos para poder aplicarlo de manera inmediata.

La finalidad detrás de *Wollok* es formar en competencias, habilidades y conceptos de suma importancia en la industria del desarrollo de software, con una propuesta que prioriza los conceptos, el razonamiento y los fundamentos necesarios para la formación de un profesional de calidad para desempeño en la industria. Partiendo de estas ideas, sin dudas que los caminos son numerosos, diferentes y que las herramientas disponibles en el mundo de sistemas son variadas. Por un lado, lo que generalmente se hace -y quienes actualmente utilizan *Wollok* también antes lo hicieron con otros lenguajes- es habiendo elegido un lenguaje de programación de uso profesional, adaptar el recorrido temático en la medida que dicha herramienta lo permita, con las limitaciones o complicaciones que ello conlleva. En un abanico de lenguajes -y a riesgo de simplificar en exceso- que va desde un Smalltalk, más puro de objetos y menos comercial hasta un ampliamente difundido JavaScript con conflictiva identificación con el paradigma de objetos, pasando por Ruby, Java, C++ o Python, por nombrar algunos, la elección del lenguaje está dada por encontrar un equilibrio razonable entre diversos factores: algunos más pedagógicos y otros más profesionales, sin ignorar otros planos que conforman el contexto de decisión, como el deslumbramiento o recelo a las nuevas tendencias, la disponibilidad de documentación, los conocimientos previos de los docentes, las inercias institucionales, etc. Otra alternativa, que es la que dio origen a este lenguaje, es partiendo de una cierta selección, ponderación y organización de una propuesta educativa, construir un lenguaje que la sostenga, acompañe y facilite. Retomando lo anterior, el equilibrio pretendido surge de ponderar fuertemente el aspecto didáctico por sobre otros, con un guiño a las nuevas tendencias en materia de desarrollo de software que *Wollok* implementa.

Por otra parte, pensando la enseñanza de la programación como parte de una formación integral que pretende que toda persona adquiera formas de modelar la realidad, pueda

desarrollar un pensamiento abstracto y sea capaz de apropiarse de la tecnología que lo rodea, *Wollok* propone un espacio fecundo y a la vez contenido para la experimentación, el juego y desarrollar el potencial creativo y creador de los estudiantes. Si bien está aún muy lejos herramientas educativas como puede ser Scratch o Pilas Bloques, por citar algunas que tienen un notable esfuerzo de acercamiento al lenguaje y a la psicología infantil, podría analizarse la pertinencia de utilizarla con estudiantes con un poco más de experiencia, no en programación estrictamente, pero sí en utilización de software, con más autonomías, con mayor capacidad de lectura e interpretación. En este sentido, lo que aporta *Wollok* de diferente es la metáfora de los objetos que se envían mensajes como forma de representar la realidad -que es la esencia del paradigma de objetos- de una forma más accesible y sencilla que lo que tal vez se imagina quien conoce los lenguajes profesionales de objetos y no quiere eso para sus estudiantes.

Los resultados reflejan que los colectivos docentes estudiados han encontrado en la herramienta una respuesta satisfactoria para sus búsquedas educativas y allí radica su motivación para continuar utilizando *Wollok* y mejorándolo. Su posible extrapolación a otros contextos educativos o eventualmente su apropiación por parte de otros docentes e instituciones educativas, si bien obviamente está mediada por la posibilidad de conocer la herramienta y tener algún antecedente favorable comprobable como la que este trabajo presenta, dependerá principalmente de cuáles son las preguntas pedagógicas que dichos sujetos construyan y cómo interpretan la realidad de sus estudiantes y lo que buscan para ellos.

Referencias

[ACS02] E. Allen, R. Cartwright, and B. Stoler. Drjava: A lightweight pedagogic environment for java. In ACM SIGCSE Bulletin, volume 34, pages 137–141. ACM, 2002.

[BC04] J. Bennedsen and M. E. Caspersen. Teaching object-oriented programming – towards teaching a systematic programming process. In Eighth Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts. Affiliated with 18th European Conference on Object-Oriented Programming (ECOOP 2004), 2004.

[CO16] W. Cazzola and D. M. Olivares. Gradually learning programming supported by a growable programming language. IEEE Transactions on Emerging Topics in Computing, 4(3):404–415, 2016.

[GF03] K. E. Gray and M. Flatt. Professorj: a gradual introduction to java through language levels. In Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, pages 170–177. ACM, 2003.

[GLPDP11] Griggio, C. Leiva, G. Polito, G. Decuzzi, G. Passerini, N. “A programming environment supporting an prototype-based introduction to OOP”. European Smalltalk User Group, 2011.

[Jen02] T. Jenkins. On the difficulty of learning to program. In Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences, volume 4, pages 53–58. Citeseer, 2002.

[LP07] Lombardi, Carlos. Passerini, Nicolás. Cesario, Leonardo. "Instancias y clases en la introducción a la programación orientada a objetos". Smalltalks 2007 – Primera Conferencia Argentina de Smalltalk, 2007.

[KQPR03] M. Kölling, B. Quig, A. Patterson, and J. Rosenberg. The bluej system and its pedagogy. Computer Science Education, 13(4):249–268, 2003.

[PFTD17] Passerini, Nicolás. Lombardi, Carlos. Fernandes, Javier. Tesone, Pablo. Dodino, Fernando. "Wollok: Language+ IDE for a gentle and industry-aware introduction to OOP." 2017 Twelfth Latin American Conference on Learning Technologies (LACLO). IEEE, 2017.

[PFTF15] Passerini, Nicolás. Fernandes, Javier. Tesone, Pablo. Fortini, Débora. “Wollok - Relearning How To Teach Object-Oriented Programming”. CONAIISI 2015.

[SGURGH15] J. E. Sánchez-García, M. Urías-Ruiz, and B. E. Gutiérrez-Herrera. Análisis de los problemas de aprendizaje de la programación orientada a objetos. Ra Ximhai, 11(4):148–175, 2015.

[SP13] Spigariol, Lucas. Passerini, Nicolás. “Enseñando a programar en la orientación a objetos”. CONAIISI, 2013

[Spi15] Spigariol, Lucas (2015) “Estrategias pedagógicas para la enseñanza de la programación”. Tesis de Magister en Docencia Universitaria. UTN FRBA.

[Spi16] SPIGARIOL, LUCAS (2016) IEEE Congreso Argentino de Ciencias de la Informática y Desarrollos de investigación. (CACIDI). UNSAM/Universidad CAECE/Universidad Central de Chile. Argentina. Diciembre 2016. "A pedagogical proposal for teaching object-oriented programming: Implementation through the educational software Wollok," Publicado: Electronic ISBN: 978-1-5090-2938-9

[Uys] M. P. Uysal. The effects of objects-first and objects-late methods on achievements of oop learners.

Sitio oficial de Wollok: https://www.wollok.org/

Repositorio del proyecto: https://github.com/uqbar-project/wollok-language

Repositorios de ejemplos de código: https://github.com/wollok

Fundación Uqbar: http://www.uqbar.org/

Estrategias de enseñanza de la programación y su aplicación en aulas con estudiantes con discapacidad

María Fernanda Golobisky[1], Rosana Portillo[2]

Resumen: El presente trabajo propone comentar las prácticas de enseñanza y su didáctica en el planteo de ejercicios de programación a estudiantes con algún tipo de discapacidad, utilizando la metodología de enseñanza y aprendizaje basada en indagación. La didáctica de clases propuestas por los docentes de escuelas especiales presenta una forma de ayudar a los estudiantes a que resuelvan diferentes problemas planteando una estrategia de resolución y que puedan comprender de manera más acabada la función de cada actividad en el total. Asimismo, les provee herramientas conceptuales que les permiten realizar diferentes observaciones, conclusiones que colaboran en ampliar sus procesos de abstracción y fortalecer el pensamiento computacional en la resolución de problemas. A pesar de que el aprendizaje de la programación requiere de una práctica progresiva e intensiva para comprender distintos conceptos, en las diferentes aulas de programación en escuelas especiales se estimula el manejo de la computadora como una herramienta pedagógica y de rehabilitación. La atención de los alumnos es individualizada o en grupos pequeños, en virtud de sus potencialidades, nivel de independencia, ritmo de trabajo y edad. En este artículo se relatan experiencias de clases desarrolladas por docentes de escuelas especiales que han realizado el curso "La programación y su didáctica 1" dictado en 2018 en el marco del convenio entre la Fundación Sadosky y la UTN Facultad Regional Santa Fe.

Palabras claves: Didáctica de la Programación, discapacidad, accesibilidad, aprendizaje por indagación.

1 Departamento de Sistemas, Facultad Regional Santa Fe, Universidad Tecnológica Nacional mfgolo@santafe-conicet.gov.ar

2 Departamento de Sistemas, Facultad Regional Santa Fe, Universidad Tecnológica Nacional rportill@frsf.utn.edu.ar

Introducción

La educación implica el ejercicio de prácticas específicas que se realizan en determinados espacios institucionales, cuya organización y objetivos garantizan el sistema de relaciones subjetivas que la hacen posible. La práctica educativa se especifica a partir de la enseñanza. No hay práctica docente sin enseñanza, ya que ella organiza la escena en la que se vinculan los sujetos a través del conocimiento. Parafraseando a Litwin [1] se entiende que investigar las diferentes prácticas es mirarlas en las condiciones naturales y en los contextos en que se desenvuelven, lo que redunda en una mirada privilegiada para aquel que se interroga a partir de reconocer e indagar su propio accionar. Con esta premisa en mente es que durante el segundo semestre del 2018 se desarrolló, en la UTN Facultad Regional Santa Fe (UTN-FRSF), una nueva capacitación docente dictada en el marco del convenio con la Fundación Sadosky. Los docentes asistentes pertenecían a diferentes niveles educativos (inicial, primario y secundario), destacándose en esta oportunidad la asistencia de docentes de escuelas especiales, cuyas aulas alojan a niños con diferentes discapacidades. La discapacidad forma parte de la condición humana. Todas las personas tenemos capacidades y necesidades diferentes, pues todos somos diferentes, y dentro de esa diversidad se incluyen las personas con discapacidad. De este modo, debemos entender que la discapacidad es una diferencia más entre otras. Y esa diferencia es una condición permanente que acompaña a la persona.

Existen diferentes tipos de discapacidades entre las que podemos mencionar:

Discapacidad intelectual (DI): Es aquella que presenta una serie de limitaciones en las habilidades diarias (aprender, comprender y comunicarse) que una persona aprende y le sirven para responder a distintas situaciones en la vida. Según [2] la DI se clasifica en los siguientes niveles, de acuerdo al coeficiente intelectual (CI) de la persona (Tabla 1):

Tabla 1. Clasificación de los niveles de DI.

Nivel de DI	CI
Leve	50-55 a 70
Moderada	35-40 a 50-55
Grave	20-25 a 35-40
Profunda	20-25

La discapacidad intelectual generalmente es permanente, es decir, para toda la vida. Es importante señalar que la DI no es una enfermedad mental sino del neurodesarrollo, es un trastorno con base neurológica que puede afectar la adquisición, retención o aplicación de habilidades específicas o conjuntos de información. Consiste en alteraciones en la atención, la memoria, la percepción, el lenguaje, la resolución de problemas o la interacción social. Por ello, a las personas con discapacidad intelectual les cuesta más que a los demás aprender, comprender y comunicarse.

Discapacidad Física o Motriz (DF): Este tipo de discapacidad implica una disminución de la movilidad total o parcial de uno o más miembros del cuerpo, la cual dificulta la realización de actividades motoras convencionales. Se pueden encontrar diferentes tipos de discapacidad física:

- Según su causa:
    - Motrices con o sin afectación cerebral
    - Debidas a enfermedad
    - Mixtas
- Según la zona afectada:
    - Discapacidad motriz de las extremidades inferiores
    - Discapacidad motriz de las extremidades superiores, tronco, cuello y cara
    - Otras discapacidades motrices

Discapacidad Auditiva (DA): Es un déficit total o parcial en la percepción de la audición que se evalúa por el grado de pérdida de la audición en cada oído. Las personas con esta discapacidad se distinguen entre:

- *sordas*, poseen una deficiencia total o profunda
- *hipoacúsicas*, poseen una deficiencia parcial, es decir, que cuentan con un resto auditivo.

La discapacidad auditiva no viene acompañada necesariamente de otra discapacidad, por lo que las personas sordas no tienen un intelecto menor.

Discapacidad Visual (DV): Es la disminución parcial o total de la vista. De acuerdo al grado de limitación de la visión se distingue entre personas ciegas y personas con disminución visual. La pérdida grave de funcionalidad de la visión se va a manifestar, por un lado, en limitaciones muy severas de la persona para llevar a cabo de forma autónoma sus desplazamientos, las actividades de vida diaria, o el acceso a la información. Por otro, en restricciones para el acceso y la participación de la persona en sus diferentes entornos vitales: educación, trabajo, ocio, etc., y que adoptan la forma, no sólo de barreras físicas y arquitectónicas, sino también sociales y actitudinales.

La Convención Internacional sobre los Derechos de las Personas con Discapacidad [3] establece que “la discapacidad es un concepto que evoluciona y que resulta de la interacción entre las personas con deficiencias y las barreras debidas a la actitud y al entorno que evitan la participación plena y efectiva en la sociedad, en igualdad de condiciones con los demás”. Discapacidad es lo que ocurre cuando las necesidades funcionales de una persona no son tenidas en cuenta por el entorno físico y social en el que vive, poniéndola en una situación de desventaja e inequidad, que se convierte en una responsabilidad social, en la que todos estamos involucrados. Y es justamente debido a esa responsabilidad que los docentes de escuelas especiales participaron de la capacitación en busca de estrategias y propuestas didácticas que les permitan realizar una educación

inclusiva, incorporando las tecnologías de información y comunicación (TIC) en el trabajo del aula, buscando recursos que promuevan aportes significativos para cada tipo de discapacidad y mejoren la calidad educativa de sus estudiantes.

Radenski en [4] estableció que la programación es considerada por muchas personas una actividad ardua y destinada sólo a un pequeño grupo de la población. En [5] Resnick et al., señalaron que los primeros lenguajes de programación eran muy difíciles de usar y muchos estudiantes no podían aprender su sintaxis; más aún, la programación era introducida con actividades que no resultaban interesantes ni atractivas a los jóvenes, ya que no estaban conectadas con sus intereses y experiencias.

En este artículo reflexionaremos sobre la enseñanza de la didáctica de la programación a estudiantes con discapacidad.

Metodología de trabajo

Esta experiencia de la enseñanza de la programación a estudiantes con discapacidades varias se desarrolló en dos escuelas de la ciudad de Santa Fe. En el primer establecimiento[3] se trabajó con alumnos sordos e hipoacúsicos, mientras que en el segundo[4] con alumnos con DI y DF (chicos en sillas de ruedas, con parálisis cerebral con movilidad reducida, con baja visión, con problemas en el habla y en la audición, con síndrome de down, y con autismo).

Experiencia 1:
Fue realizada en un 5° grado al que asisten alumnos que si bien no presentan DI, tienen una deficiencia en el lenguaje debido a la DA que poseen, y que afecta el nivel de abstracción de su pensamiento. La experiencia consistió en un taller de 5 días con actividades de 1 hora de duración diaria. Los objetivos perseguidos fueron: i) interpretar el concepto de programa (como una secuencia de comandos); ii) interpretar los conceptos básicos de un programa (autómata, comandos, repeticiones simples); iii) relacionar estos conceptos con los de la herramienta Pilas Bloques; iv) incorporar el vocabulario de programación a la tarea a realizar; v) resolver problemas y situaciones planteadas a través de juegos-actividades. En la Tabla 2 se describen las actividades iniciales planificadas por la docente y a realizarse sin computadora, mientras que en la Tabla 3 se visualiza una descripción de actividades posteriores, algunas con y otras sin computadora.

3 Escuela Especial para Niños Sordos e Hipoacúsicos N°2012 “Gral. San Martín”.
4 Escuela Especial Nº 2009 “Dr. Bernardo de Monteagudo”.

Tabla 2. Actividades iniciales.

Actividades Iniciales – sin computadora -
• Se les solicita a los estudiantes observar tarjetas con diferentes imágenes de robots y videos cortos. Atender a las preguntas de la docente, tales como: *¿qué es esto? ¿saben cómo se llama?* • Se les pide atender al nombre: Robot, por lectoescritura, por lectura labial, y dactilológico[5]. Y posteriormente el grupo debe responder a la pregunta *¿Cómo se llama?* • Se da a conocer la seña que identifique la palabra robot. A continuación, se plantean nuevas preguntas tales como: *El robot; ¿camina sólo?, ¿mueve las manos sólo? ¿gira sólo?, ¿piensa?, ¿qué les parece?, ¿o alguien tiene que darle la orden?*, se entabla un diálogo con la docente. Se invita a los alumnos a jugar a los robots: prestando atención a diferentes órdenes presentadas en tarjetas, leerlas, reconocerlas y por turnos ejecutarlas. Armar las secuencias con las tarjetas (no más de dos) y pegarlas en el pizarrón. Existe intercambio de roles y se refuerza la indagación en grupo. Finalmente, se comienza con la explicación de analogías entre secuencia de órdenes y procedimiento, entre robot y autómata. • A continuación, se los invita a jugar a ser autómatas. El juego consiste en ejecutar órdenes para "avanzar" presentadas en tarjetas y una grilla simple dibujada con tiza en el piso. El alumno debe ubicarse al inicio de la grilla y avanzar de acuerdo a la cantidad que se indique (una tarjeta -un casillero, dos tarjetas -dos casilleros, y así sucesivamente). Luego, deben cambiar de roles. Dentro del juego se observan las palabras a lo que se llamará Autómata, Primitivas y Procedimiento por lectura ideovisual[6], por lectura labial y dactilológico. Se construye entre todos una seña que identifique dichas palabras.

Tabla 3. Descripción de las actividades.

Actividad lúdica - sin computadora -
• Se realiza la presentación de un auto a control remoto y se comienza a jugar libremente. • La docente comienza con la indagación planteando preguntas tales como: *¿el auto se mueve solo? ¿yo tengo que guiarlo?*, etc. • El juego se realiza a través de una grilla en el piso y se comienza pegando las órdenes en el pizarrón. Se agregan casilleros a la grilla y nuevas tarjetas con primitivas. En grupo, se piensa y se desarrolla la estrategia para lograr el movimiento del auto según primitivas. La docente indaga acerca de la secuencia de órdenes. Se observa la repetición de la misma orden o primitiva y el uso de la

5 Sistema de comunicación que transmite información mediante el uso de los dedos de la mano.

6 Los lectores leían ideas y no símbolos gráficos y estas ideas estaban relacionadas con algo más allá de los mismos símbolos (Decroly, Bélgica 1936).

palabra Repetir, de acuerdo al número de veces que se repite. • El grupo debe responder a las preguntas del docente: *¿cuál es el autómata?, ¿cuáles son las órdenes? ¿cómo llamamos a las órdenes? ¿cómo se llama a toda la secuencia de comandos?* Las respuestas se apoyarán en tarjetas con las respectivas palabras: autómata, primitivas, procedimiento.
Actividad utilizando Pilas Bloques
• Al inicio se recuerdan los conceptos dados en clases anteriores y se realiza la presentación de la aplicación Pilas Bloques. La docente explica en qué consiste el juego y se comienza por el primer desafío "El Alien toca el botón", para su desarrollo se conecta la netbook al proyector y entre todos resuelven el problema y corrigen la actividad. En base a preguntas, que realiza la docente, se identifican las primitivas, se las lee y señalan en la pizarra. • Las mismas acciones se plantean para el desafío Nro. 2 "El gato en la calle" y el Nro. 3 "No me canso de saltar".

Experiencia 2:
Las docentes tenían dos grupos de estudiantes:

- estudiantes con DI leve a moderada, que contaban con ciertas habilidades cognitivas que les permitió desarrollar las actividades planteadas, sin limitaciones ya que se trabajó fuertemente con la figura del compañero tutor, con el apoyo, la orientación y la guía de cómo ir desarrollando las actividades
- alumnos con DI profunda con muchos impedimentos físicos asociados: en el desarrollo del lenguaje e interpretación de consignas (debiendo trabajar con imágenes, flechas, etc.); estudiantes con movilidad reducida o en silla de ruedas (por lo que se planificaron actividades en el piso para que éstos pudieran movilizarse con sus sillas, o que las docentes pudieran trasladarlos).

En ambos grupos la actividad propuesta fue sin computadoras, del tipo desenchufada, con el objetivo de ofrecer a los estudiantes una experiencia lúdica, vivencial, en la cual pudieran aprender conceptos básicos de programación, tales como autómatas y comandos. El tema elegido fue uno trabajado a lo largo del año: alimentación saludable. Para ello, utilizaron una rayuela dibujada en el piso e imágenes móviles (pictogramas[7]) que los estudiantes, junto con las docentes, fueron identificando y dialogando sobre lo que representaban. En la tabla 4 se observa una descripción de las actividades.

7 Signos icónicos dibujados y no lingüísticos, que representan esquemáticamente un símbolo, objeto real o figura. Se emplean en los tratamientos para personas con autismo, como una herramienta de comunicación y organización.

Tabla 4. Descripción de las actividades.

• *Introducción:* Se comentó a los estudiantes sobre el juego a realizar. • *Dar a conocer el espacio de juego:* se identificaron las imágenes presentes en la rayuela, dialogando sobre si eran o no alimentos saludables. • *Aproximación a los conceptos:* se invitó a los estudiantes a conocer imágenes de diferentes "autómatas": las observaron, se indagó sobre si los vieron en otra oportunidad, si tienen uno en casa, aprendieron el nombre de cada uno (robot, máquina, etc.), discutieron sobre ellos *(¿caminan, se alimentan, reconocen alimentos saludables?*). Luego, se les propuso elegir a un compañero como autómata del juego y asignarle una de las imágenes vistas, que llevó consigo a modo de insignia. • *Presentación del enunciado:* se explicó la actividad a realizar (un bebé debe desplazarse por la rayuela alimentándose de manera saludable para crecer), y se incluyeron en la rayuela imágenes sobre el estado inicial (bebé) y final (niño sano) del autómata. • *Trabajo colaborativo:* se explicó a la clase que el autómata solo podía cumplir la consigna con la colaboración de todos ellos, quienes debieron indicarle, con el uso de comandos, qué y cómo cumplir el objetivo. • *Presentación de pictogramas:* se les presentaron tarjetas con las primitivas disponibles (avanzar, mover a la derecha, etc.) y una docente ejemplificó cada acción. • *Estrategia de resolución:* antes de que el autómata inicie su recorrido, se les pidió a los compañeros que entre todos armen con los pictogramas la secuencia de acciones que éste debía realizar.

Principales Resultados

En este artículo se sintetiza el trabajo realizado por docentes de escuelas especiales poniendo en práctica diversas estrategias para enseñar a programar a estudiantes con diferentes discapacidades. En virtud de las potencialidades, el nivel de independencia, el ritmo de trabajo y la edad de los alumnos se realizó una adecuación de las actividades, seleccionando y acotando el contenido, y graduando el nivel de especificidad de los temas impartidos. Como resultado de la experiencia desarrollada podemos concluir que el carácter lúdico y vivencial de las propuestas entusiasmó a los estudiantes permitiéndoles aprender de una forma más atractiva y divertida y construir conocimientos logrando un aprendizaje más autónomo. El uso pedagógico y significativo de las TIC (tales como la herramienta Pilas Bloques, la pizarra digital, las netbooks) los motivó ampliamente y les permitió la apropiación de nuevas competencias multitarea, participación, resolución de problemas, producción, aprendizaje colaborativo. Las intervenciones docentes, las analogías planteadas, las orientaciones realizadas a través de una indagación estructurada, donde el docente provee la pregunta inicial y delinea el procedimiento general de la solución, ayudó a los alumnos a sacar conclusiones y resolver los problemas presentados.

Como conclusión general, la experiencia resultó satisfactoria dado que en los alumnos con DA, el aprendizaje mediante indagación los ayudó a incorporar un pensamiento computacional -trabajando la memoria, la atención, el seguimiento de instrucciones, la división del problema en subtareas- posibilitándoles organizarse mentalmente, logrando la abstracción del pensamiento y estimulando su percepción auditiva por medio de actividades interactivas. Si bien esta experiencia constituyó un primer acercamiento con elementos del pensamiento computacional, para que estos alumnos puedan aprehender estos conocimientos deberían continuarlos trabajando en el futuro.

Con respecto a los estudiantes con DI, estos aprenden cada uno a su ritmo, es decir, no todos al mismo tiempo. Realizar actividades como la de aprender a programar, de manera regular (más sostenida en el tiempo), podría llegar a ser beneficiosa para ellos ya que les permitiría, en algún momento, alcanzar el pensamiento computacional. Con la propuesta didáctica del taller (participación colaborativa a través de un juego), además del proceso de indagación, en función de su nivel cognitivo, pudieron resolver problemas mediante el planteo previo de una estrategia. Asimismo, lograron introducir los conceptos básicos de programación como la secuencia de actividades, trabajando en funciones cognitivas como la atención y el seguimiento de instrucciones.

Como conclusión podemos decir que es innegable el potencial de esta actividad. Teniendo en cuenta que estos alumnos nunca habían tenido contacto previo con este tipo de experiencias, esta actividad se constituyó en un disparador para la realización de futuros talleres.

Referencias

[1] Litwin, Edith (2008). El oficio de enseñar. Condiciones y contextos. Buenos Aires: Paidós.
[2] Suplemento del Manual diagnóstico y estadístico de trastornos mentales, quinta edición. DSM-5® septiembre 2016. Disponible en: https://dsm.psychiatryonline.org/pb-assets/dsm/update/Spanish_DSM5Update2016.pdf.
[3] Art. 1 de la Convención Internacional sobre los Derechos de las Personas con Discapacidad (Aprobado en 2006 por la Asamblea General de las Naciones Unidas y en Argentina en 2008 a través de la Ley 26.378). https://www.un.org/esa/socdev/enable/documents/tccconvs.pdf
[4] Radenski, A. (2006). Python First: A lab-based digital introduction to computer Science. ITiCSE '06 11th Annual Conference on Innovation and Technology in Computer Science Education Bologna, Italy - June 26 - 28, 2006. http://dl.acm.org/citation.cfm?id=1140124&picked=prox
[5] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Siver, J., Silverman, B., y Kafay, Y. (2009). Scratch: Programming for all. Communications of the ACM, 52 (1), 60-67. DOI:10.1145/1592761.1592779.

SECCIÓN N°2

FORMACIÓN DOCENTE

Hacia una didáctica de la programación para la secundaria argentina

Pablo E. "Fidel" Martínez López[1], Alfredo Sanzo[2], Fernando Schapachnik[3]

Resumen: Dentro del marco de la iniciativa Program.AR, la Fundación Sadosky desarrolló un curso para la educación secundaria denominado *"La programación y su didáctica"* (LPYSDI), el cual se viene dictando con éxito desde 2015 a docentes de todo el país a través de un modelo de convenios con Universidades Nacionales. En este artículo presentamos el diseño de la segunda parte de este curso, denominado *"La programación y su didáctica"* (LPYSD2) concentrándonos en los objetivos perseguidos, la metodología utilizada, el marco conceptual considerado y las características del curso resultante. Este curso también se comenzó a dictar este año a través del modelo de convenios mencionado y así se integra a los esfuerzos dentro de la estrategia de la iniciativa Program.AR en pos de conseguir que la materia Ciencias de la Computación sea dictada en todos los niveles educativos del país como cultura general.

1 Universidad Nacional de Quilmes Fundación Dr. Manuel Sadosky. fidel©unq.edu.ar
2 Fundación Dr. Manuel Sadosky, Universidad de Buenos Aires, FCEyN. asanzo@fundacionsadosky.org.ar
3 Partially supported by UBACyT 20020130200032BA. Fundación Dr. Manuel Sadosky, Universidad de Buenos Aires, FCEyN. fschapachnik©fundacionsadosky.org.ar

1. Introducción

En el marco de la iniciativa Program.AR[4], la Fundación Sadosky diseñó y dictó un curso que llamaremos LPYSD1 (*"La programación y su didáctica, parte 1")* [FSO15]. Este curso fue concebido para su dictado a estudiantes de secundaria, y se realizaron diversas capacitaciones a docentes de secundaria para que puedan trasladarlo al aula. Desde 2014 la Fundación Sadosky capacita equipos universitarios de todo el país para que dicten dichos cursos de formación docente en programación con este material. Estos equipos son asignados mediante un proceso de selección determinado por un jurado internacional, y ya han formado a más de 1500 docentes. LPYSD1 es una primera introducción a la programación, y a una forma de enseñarla, y tiene varias características innovadoras. Sin embargo, el curso deja varias cuestiones importantes sin atender.

En este trabajo explicaremos el diseño y prueba de la continuación de LPYSD1, que llamaremos LPYSD2 (*"La programación y su didáctica, parte 2'*).

2. Planificación de la continuación

2.1. Objetivos a cumplir

El primer paso cuando nos planteamos diseñar e implementar LPYSD2 fue establecer cuáles eran los objetivos que buscábamos con tal curso. Nos planteamos como objetivos que el curso incluyese:

- Procesar/computar datos.
- Abordar problemas tecnológicos/informáticos relacionados a la experiencia cotidiana.
- Proveer al docente confianza sobre los temas que imparte.
- Incluir problemas no triviales para que los cursantes resuelvan.
- Codificar, poner en práctica el conocimiento, concretizar.

Estos temas fueron elegidos por las razones que se exponen a continuación. Este curso se basa en la experiencia de los autores sobre los temas a incluir en un curso de programación que sea continuación de un curso inicial (ver por ejemplo [SP88, AHU83]).

Procesar datos, porque en LPYSD1 no se incluyó la idea de un programa como procesador de información, lo cual creemos es central en la concepción de la programación. En esa primera parte la forma de concebir a los programas era como meras series de instrucciones para que un autómata haga algún cambio en el escenario. En esta segunda parte (LPYSD2; se incluye la concepción de que un programa puede también recibir una entrada y producir una salida.

4 http://program.ar

El segundo ítem tiene que ver con el objetivo último de la iniciativa Program.AR sobre explicar las características de las soluciones tecnológicas que nos rodean hoy en día. El primer curso, al ser introductorio, trabajó sobre ejemplos extremadamente simples, y no estaba presente ese puente entre el qué y el cómo en las soluciones tecnológicas logradas con programación. Se busca con esta segunda parte del curso explicitar que con cierto algoritmo o con cierta estructura podría resolverse un problema con el que los cursantes están en contacto cotidianamente. También que puedan imaginarse cómo funcionan los programas con los que interactúan a diario, al menos en un nivel básico. Por ejemplo, formular hipótesis sobre cómo funciona un predictor de texto de un celular, o cómo encuentra Google lo buscado, etc. La inclusión de lecturas complementarias es importante también para conseguir este objetivo.

El tercer ítem está relacionado al hecho de que en el primer curso todo el contenido estaba destinado a que el docente lo usase en los cursos para sus alumnos, y hasta ahí llegaba su saber. En este curso buscamos entonces que los cursantes puedan conocer más que lo que deben dictar en un curso, y de esta forma manejarse mejor en el aula. Es decir, proveerles una serie conocimientos disciplinares adicionales al curso anterior con el objetivo de generar confianza. Gracias a esto, el cursante puede entender mejor el rol que tuvo y tiene LPYSD1 en la enseñan a básica de la programación, sin concebirlo como una sobresimplificación, y entendiendo mejor algunas de las características de aquel. El cuarto ítem, que los ejercicios sean no triviales, se puso como objetivo para construir actividades desafiantes y enriquecedoras, que permitan por un lado lograr la satisfacción de haber resuelto un problema interesante y por el otro mostrar la complejidad inherente de algunos de los problemas computacionales más comunes.

El último ítem viene de la convicción de que en didáctica, la apropiación del conocimiento es mejor cuando el cursante se ejercita y produce resultados concretos. En la temática de la programación esto significa escribir código y hacerlo

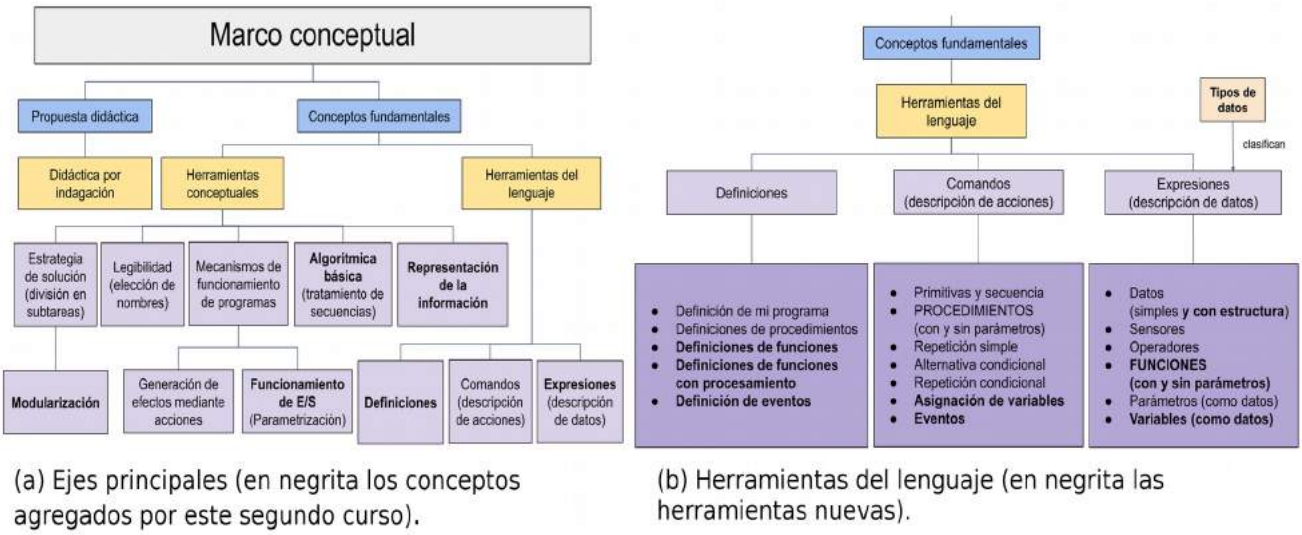


Figura 1. Marco conceptual de LPYSD2

funcionar. Además, si producen alguna aplicación, por sencilla que sea, el cursante sentirá que se llevó algo valioso.

Un objetivo implícito por provenir de LPYSD1, pero que vale la pena mencionar aquí, es reiterar la características del curso 1 de trabajar con entornos y actividades diversas, con características diferentes aunque una base común.

Mientras que los cursos tradicionales son *monolenguaje*, i.e. trabajan con un único entorno/lenguaje de programación (por ejemplo, cursos basados en Scratch [ML07, LJ11], Alice [Her07], Karel the Robot [Pat81], etc.), creemos que un curso que se focalice en conceptos fundamentales debe ofrecer actividades de más de una naturaleza (e.g. con computadoras y sin computadoras) y en más de un entorno/lenguaje, como una manera de mostrar y practicar el hecho de que los conceptos no son exclusivos de un tipo de actividad, de un lenguaje específico o de un entorno de trabajo determinado.

2.2. Extensión del marco conceptual

El marco conceptual de LPYSD1 se basó en 3 ejes: didáctica, herramientas conceptuales y herramientas del lenguaje. En este curso se plantea continuar con esos 3 ejes, y extenderlos acorde a los objetivos planteados.

La decisión de mantener el eje de la didáctica, que es la didáctica por indagación [Dos15], implica articular el curso alrededor de actividades que plantean un problema, invitan a resolverlo, a experimentarlo, y luego dan la teoría necesaria, en lugar de hacer la presentación teórica primero y la práctica luego. Una característica fundamental de esta propuesta didáctica consiste en que los cursantes experimenten esta forma de aprendizaje durante la capacitación, y luego reflexionen sobre la misma, con el objetivo de poder producir actividades propias que sigan estos principios de trabajo. De esta forma, ambos cursos se basan no solamente en que los estudiantes, destinatarios finales, aprendan mediante indagación, sino que los docentes, al capacitarse, también la experimenten, y luego la analicen y comprendan en profundidad.

En el eje de las herramientas conceptuales, los conceptos principales de LPYSD1, siendo tan fundamentales, también deben estar presentes: la determinación y explicitación de la estrategia al ofrecer la solución de un problema, y el hecho de que el programa debe ser legible por las personas. A ello se agregan los siguientes conceptos (ver la Fig. 1a):

- las dos formas de concebir los programas: la ya trabajada de producir ciertos efectos sobre el estado a través de acciones, y la nueva de transformar información a través de su procesamiento (a la que llamaremos “funcionamiento de Entrada/Salida”),

- la idea de que la información se representa de diversas formas dentro de un sistema de cómputo (representación de la información),
- la noción de algoritmo, su práctica a través de un algoritmo de recorrido de secuencias [SP88] y la sugerencia de que existen diversidad de algoritmos (en particular, algoritmos de ordenamiento y de búsqueda binaria, siempre acorde a los objetivos), y
- la modularización como forma de transformar porciones de código en un producto reutilizable para resolver nuevos problemas.

Finalmente, en el eje de las herramientas del lenguaje, que en el curso LPYSD1 estaban fuertemente concentradas en los comandos, el foco se trasladará al universo de las expresiones, dado que las mismas son la forma en que se describe la información. También se distinguirá la noción de definición de operaciones (procedimientos, funciones, eventos, etc.), de la noción de utilización de los mismos (ver la Fig. 1b), que en el curso LPYSD1 no se distinguían de forma explícita. Dentro del universo de las expresiones se propone trabajar con la idea de función (similar a la idea de procedimiento en el universo de los comandos), la noción de datos con estructura (como iniciación a las estructuras de datos, pero de forma simple), y profundizar en las nociones de operadores, de parámetros y variables, para lo cual se hará necesario incluir la idea de tipos de datos.

Se espera que a través de la comprensión de estos conceptos y herramientas se puedan explicar varias características de las aplicaciones modernas. La elección de estos conceptos implicó el desplazamiento de otros posibles. Por ejemplo, quedaron afuera la manipulación y definición de estructuras de datos más complejas, o nociones de aplicaciones prácticas como bases de datos, procesamiento distribuido o inteligencia artificial. Si bien los contemplamos, no encontramos la forma de incluirlos dentro del tiempo que le podíamos dedicar a cada tema sin perder precisión o riqueza.

Las herramientas conceptuales y las herramientas del lenguaje utilizadas tanto en LPYSD1 como LPYSD2 están basados en trabajos anteriores [MLBSO12, ML13] sobre las cuales el equipo de Program.AR en su manual [FSO15] hizo una selección y agregó el trabajo con bloques, con actividades unplugged, con desafíos concretos y la didáctica por indagación. Con el pasar del tiempo y las consecuentes revisiones se consolidó un material [MLCAP17] sobre el cual se continuó construyendo, agregando temas como representación de información, de tratamiento de datos con estructura (fundamentalmente secuencias) y de la exploración de la riqueza del mundo de las expresiones. Estos temas, con algunos desarrollos previos aún no documentados, fueron enriquecidos y desarrollados para abarcar los objetivos aquí propuestos.

3. El curso “La programación y su didáctica 2”

3.1 Características de LPYSD2

A la hora de diseñar un curso de esta naturaleza, es importante tomar algunas decisiones desde el comienzo, que luego afectarán las características del curso en sí. La primera de las decisiones ya había sido tomada, y es que este curso es una continuación de LPYSD1, por lo que los contenidos dados en ese curso son prerrequisito para tomar este, y el formato de trabajo y la cantidad de horas debe ser similar al de LPYSD1: 70 hs. presenciales y 30 hs. no presenciales. Luego de eso, la más importante de estas decisiones tiene que ver con el o los entornos de programación que se utilizarán para desarrollar las actividades, ya que los mismos estarán presentes en el día a día del curso, y afectarán el tipo de producciones que se pueden lograr. También es necesario elegir estrategias didácticas para que los conceptos más complejos, como por ejemplo la idea de datos con estructura o la noción de búsqueda binaria, aparezcan de forma sencilla e intuitiva. Con esto en mente, y teniendo en cuenta los objetivos, se deben seleccionar y secuenciar los temas, y finalmente, establecer los mecanismos de evaluación.

3.1.1. Entornos de programación

Dado que el curso LPYSD1 se basó completamente en programación en entornos con bloques, y que la programación basada en texto, si bien más adecuada profesionalmente, plantea desafíos didácticos mayores (por la complejidad de manejar la sintaxis de un lenguaje, y por la mayor facilidad para cometer errores al usar texto), decidimos desde el comienzo que los entornos de programación que propusiésemos tenían que estar basados en bloques. Luego de considerar varios entornos, finalmente elegimos dos: GobstonesWeb[5] y App Inventor[6].

El primero de ellos, diseñado por un equipo de la UNQ, permite definir actividades completas con facilidad, focalizando al estudiante en los aspectos conceptuales elegidos y eliminando otros innecesarios, incluye el tratamiento de datos con estructura, expresiones y todos los conceptos elegidos, y tiene como bonificación adicional el hecho de que, al tener acceso al código fuente del entorno, podemos realizar ajustes en la interfaz y la presentación.

El segundo, diseñado por el MIT, permite la definición de programas que funcionen en celulares, logrando de esta manera productos de software que pueden trascender su uso áulico; si bien es más difícil focalizar los conceptos, al usarlo en combinación con el primero se puede simplificar el acceso a su complejidad.

Ambos se pueden utilizar en forma *online* a través de Internet, o instalarlos para su uso local (aunque en el caso de App Inventor esto requiere un trabajo algo complejo). De esta forma, si bien el curso resulta más fácil de dictar teniendo conexión a Internet, no depende exclusivamente de ello.

5 http://gobstones.github.io/gobstones-jr
6 http://ai2.appinventor.mit.edu/

3.1.2. Estrategias didácticas

Entre las estrategias didácticas, como primer punto decidimos reutilizar la combinación de actividades que usan computadoras con otras que no las requieren, como forma de abordar los conceptos antes de pasar a las máquinas. Pero se hizo necesario buscar formas para transmitir por indagación las ideas complejas, tales como la noción de dato con estructura, la idea de que las computadoras manejan datos y pueden procesarlos, algoritmos complejos como el de búsqueda binaria, etc.

Para la noción de datos con estructura elegimos el dominio de las cartas españolas. Una carta es un dato con estructura que posee dos partes bien diferenciadas, y sin embargo es una sola entidad.

Un mazo de cartas es una instancia del tipo lista, que al ser presentado de la forma adecuada, permite poner en claro algunas de las características de las listas en los lenguajes de programación. En forma intencional omitimos la capacidad de definir nuevos tipos de datos con estructura, limitándonos solamente a las cartas y las listas (de cartas, de números y de letras), ya que el tema de diseño de otras estructuras es mucho más complejo conceptualmente, y no era parte de los objetivos planteados.

Para transmitir la idea de que las computadoras procesan datos, extendimos la metáfora de varias actividades sin computadoras del curso LPYSD1, donde el docente funcionaba como una computadora. En este curso, el docente-autómata no solo entiende comandos, sino que además puede responder ciertas preguntas. De esta forma, resulta más sencillo mostrar el funcionamiento de E/S de los programas, y la necesidad de contar con expresiones más ricas en el lenguaje de programación.

Finalmente, para transmitir la idea de algoritmos complejos, como los de ordenamiento o de búsqueda binaria, elegimos, para los primeros, presentar solo el ordenamiento por selección en forma modularizada y dejar el estudio de otras formas para lecturas adicionales, y para el segundo, utilizar una actividad sin computadoras diseñada por el equipo de la UNER para la especialización en docencia de computación resultado de un convenio con la Fundación Sadosky. Esta actividad, en la forma de un juego por parejas, no se llega a codificar, ya que el código necesario requiere características más avanzadas como manejo de índices o recursión, ambos fuera del alcance propuesto para este curso, pero sin embargo, consigue que los cursantes descubran por indagación la estrategia de buscar en forma binaria.

Como muestra del tipo de actividades que se proponen, elegimos las involucradas en la unidad de introducción a listas, por ser representativas del estilo propuesto: una combinación de actividades sin computadoras y con computadoras que se interrelacionan para favorecer la indagación guiada.

- Comenzamos con una actividad sin computadoras que llamamos *"Delfina y su papá van al almacén'*. En ella contamos una breve historia de una visita al almacén donde el almacenero anota precios en una listita de papel, comenzando por un sachet de leche que es lo primero que compran, y luego, planteamos que el docente funciona como un autómata que responde preguntas acerca de dicha listita. Los cursantes deben responder algunas preguntas vinculadas con la historia (tales como "¿Cuántos productos compraron en total?" o "¿Cuál fue el precio de la leche?"), para lo cual deben descubrir las operaciones de lista (primer elemento, cantidad de elementos, etc.). Las preguntas se organizan en 2 niveles, el primero con respuestas directas desde las respuestas del autómata, el segundo con la necesidad de realizar algún tipo de cuenta o procesamiento de dichas respuestas.
- Sigue una actividad con computadoras llamada *"Descubrir la lista"*[7], donde practican el uso de las operaciones recién descubiertas y observan cómo se concretan en el entorno de programación.
- Luego siguen otras dos actividades sin computadoras, *"Los mazos también son listas'* y *"Los segundos serán los primeros*", diseñadas para entender que los mazos de cartas son listas, porqué las listas se acceden de a un elemento por vez, y comprender que puede haber operaciones sobre listas que describan otras listas (en particular, la operación sinElPrimero, que describe la lista resultante de quitar el primer elemento de una lista, necesaria para realizar luego procesamientos lineales sobre las listas). En particular, en la primera de ambas, el docente vuelve a funcionar como un autómata que responde preguntas, en este caso sobre un mazo de cartas (con algunas pocas cartas, por ejemplo entre 7 y 10), pero no les muestra el mazo y solo puede responder preguntas sobre la primera y la última de las cartas; de esta forma se entiende que algunas preguntas requieran más trabajo que otras y que las computadoras no ven los elementos individuales de una lista, sino el grupo completo.
- Se completa esta unidad con 3 actividades con computadoras: *"La lista contraataca"*[8] , *"El regreso de la lista"*[9] y "*El repartir de las cartas"*[10] . en las que se trabaja fundamentalmente con la operación sinElPrimero, y se presenta el primer recorrido lineal de listas.

3.1.3. Temas del curso

Los temas elegidos para cumplir los objetivos se presentan en la Fig. 2. Para cada uno de ellos se diseñaron varias actividades que permiten presentar las ideas más importantes de cada tema a través de la didáctica por indagación. El material didáctico

7 http://bit.ly/2HIu8vT
8 http://bit.ly/2We8w3E
9 http://bit.ly/2Wj58Eu
10 http://bit.ly/2JsREAd

resultante incluye el código necesario o las fichas para llevar adelante cada una de las actividades.

1. Repaso de los conceptos de LPYSD1
2. Sensores y operadores numéricos y booleanos
3. Datos con estrucutura: cartas
4. Funciones simples (sin parámetros)
5. Tipos de datos
6. Eventos
7. Parámetros y parametrización de operaciones (procedimientos y funciones)
8. Funcionamiento de E/S
9. Variables
10. Uso de variables como acumuladores
11. Datos con estructura: listas
12. Operaciones sencillas sobre listas (totalizaciones y transformaciones)
13. Operaciones más complejas sobre listas (filtros y búsquedas)
14. Representación de la información
15. Operaciones avanzadas sobre listas (ordenamiento)
16. Algoritmo de búsqueda binaria
17. Aplicación de integración

Figura 2. Temas cubiertos en LPYSD2, por orden de aparición.

3.1.4. Evaluación

En un curso de estas características, donde lo que se busca es transmitir conceptos didácticos además de los conceptos propios de la disciplina de la programación, es necesario que la evaluación tenga en cuenta las mismas. En el curso LPYSD1 esto se consiguió requiriendo como evaluación que los cursantes dictasen al menos una clase usando los conceptos y didáctica aprendidos, y que la misma fuese supervisada por uno de los capacitadores. Sin embargo, en este curso eso no resulta factible, pues este curso presupone que los cursantes han tomado el curso anterior, y eso, al día de hoy, no es posible lograrlo con sencillez con estudiantes de secundaria. Por eso, decidimos que el curso iba a tener dos instancias de evaluación, cubriendo los aspectos esenciales: el didáctico, el conceptual y la aplicación de conceptos.
La primera de las instancias de evaluación elegidas consiste en diseñar en grupos una actividad didáctica para presentar alguno de los temas, similar a las existentes, y exponerla durante las clases del curso, como si de una actividad más del curso se tratase. Para ello se les provee a los grupos con una ficha que los guía en el diseño de actividades, focalizándolos en determinar los objetivos didácticos, los temas a transmitir y la complejidad de la solución obtenida. Durante la presentación de estas actividades, los cursantes evaluados toman el rol de docentes, y son sus pares los que les plantean observaciones sobre la pertinencia y oportunidad de las decisiones tomadas; de esta forma, se produce un análisis interesante sobre los aspectos didácticos, y un debate muy rico sobre cómo conseguir los objetivos didácticos propuestos inicialmente. Esta actividad se propone al promediar la primera mitad del curso, se diseña durante el curso, y se evalúa en dos de las clases dedicadas especialmente a ello al promediar la segunda mitad.

La segunda de las instancias es un trabajo de investigación, del que luego deben presentar un breve informe. En este trabajo, dividido en dos partes, la propuesta es elegir una aplicación cotidiana o parte de ella, proponer una hipótesis acerca de su funcionamiento que incluya los contenidos del curso (primera parte de la actividad), y

luego investigar acerca de la misma para corroborar, refinar o refutar las hipótesis planteadas, sacar conclusiones y confeccionar un informe final que contenga todo el proceso (segunda parte de la actividad). Ambas partes se presentan en dos clases diferentes cerca del final, con el objetivo de que las hipótesis sucedan antes que la investigación.

3.1.5. Material didáctico producido

Además del código necesario para llevar a cabo las actividades con computadoras[11], desarrollamos fichas para cada una de las actividades sin computadoras, diapositivas para cada una de las clases, y estamos trabajando actualmente en un manual que le permita a un docente seleccionar algunas o todas las actividades para planificar sus propias clases.

4. Resultados preliminares

Como parte del diseño del curso, equipos de dos Universidades Nacionales (la Universidad Nacional de Quilmes (UNQ) y la Universidad Nacional del NorEste (UNNE) realizaron dictados piloto durante el 2do semestre de 2018. A partir de estas experiencias, se ajustaron varias de las características del curso, diversas actividades y se refinó el material didáctico producido.

Luego desde la Fundación Sadosky se capacitó a otras 8 Universidades Nacionales (las Universidades de Buenos Aires, de Catamarca, de San Luis, del NorOeste de la provincia de Buenos Aires, de Tucumán, de Santiago del Estero, del Comahue y la Tecnológica regional Santa Fe – UBA, UNCa, UNSL, UNNOBA, UNT, UNSE, UNComa y UTN-FRSF), y las mismas comenzaron el dictado del curso durante el 1er semestre de 2019. Estos dictados están en pleno proceso mientras escribimos estas líneas.

Los cursantes que participaron de los dictados piloto del curso quedaron extremadamente satisfechos con el material ofrecido, y manifestaron su acuerdo en que los objetivos originales se cumplieron con creces. Hace falta un análisis más profundo de resultados, el cual será posible una vez finalizados los cursos que se encuentran actualmente siendo dictados.

5. Conclusiones

Uno de los grandes aprendizajes que realizamos durante la implementación de los cursos, tanto LPYSD1 como LPYSD2, es la diferencia enorme que significa para los cursantes tener un marco conceptual de referencia y reconocerlo explícitamente. De esta forma, es posible entender rápidamente cuál es el rol de una actividad

11 https://github.com/gobstones/curso-lpysd2 y https://program-ar.gitlab.io/lpysd2/ para Gobstones-Web y App Inventor respectivamente (ambos accesibles fácilmente desde dentro de los entornos correspondientes)

determinada en el conjunto, y también construir criterios para juzgar soluciones diferentes a las propuestas, además de percibir al curso como una unidad conceptual integrada. Por esa razón, durante el diseño de este curso prestamos especial atención a la construcción de un marco conceptual que fuera en todo coherente con el de LPYSD1, y que contuviese lo necesario para conseguir los objetivos buscados.

Esperamos que este curso constituya un aporte para los docentes que deciden acercarse a nuestra disciplina.

6. Agradecimientos

Al grupo docente que dicta la materia Introducción a la Programación, primera materia de las carreras Tecnicatura Universitaria en Programación Informática y Licenciatura en Informática con orientación al Desarrollo de Software en UNQ, que viene desarrollando material didáctico, ejemplos y entornos de programación que nos permitieron focalizar con rapidez las actividades en los ejes propuestos. Al grupo de extensión de UBA, quienes siempre han brindado valiosos aportes y ahora nos dieron permiso para utilizar varias actividades de su autoría. A los equipos de todas las universidades seleccionados para dictar esta segunda parte del curso por su feedback, especialmente a los de las universidades piloto (UNQ y UNNE), que permitieron iterar el material y mejorarlo. Y como siempre, al equipo de capacitación de Fundación Sadosky (Herman, Dani, Franco, Hernán, Juli, Javi, Gus, y nuestras más recientes incorporaciones, Tomás y Fernando).

Referencias

[AHU83] Alfred V. Aho, John E. Hopcroft, and Jeffrey Ullman. *Data Structures and Algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1983.

[Dos15] J. Dostál. *Inquiry-based instruction: Concept, essence, importance and contribution.* PhD thesis, Palacký University, Olomouc, Czech Republic, 2015. ISBN 978-80-244-4507-6.

[FSO15] Pablo M. Factorovich and Federico A. Sawady O'Connor. *Actividades para aprender a Program.AR.* Fundación Sadosky, 2015. In Spanish. ISBN: 978-987-27-4161-7. http://program.ar/descargas/ manual-docente-descarga-web.pdf.

[Her07] Charles W. Herbert. A*n Introduction to Programming Using Alice*. Skills, 1st edition, 2007.

[LJ11] Annette Lamb and Larry Johnson. Scratch: Computer programming for 21st century learners. *Teacher Librarian*, 38(4):64–68, April 2011.

[ML07] David J. Malan and Henry H. Leitner. Scratch for budding computer scientists. *SIGCSE Bull.*, 39(1):223–227, March 2007.

[ML13] Pablo E. Martínez López. Las Bases Conceptuales de la Programación. Una nueva forma de aprender a programar. Author, E-Book, december 2013. In Spanish. ISBN: 978-987-33-4081-9. Ebook URL:
http://www.gobstones.org/bibliografia/Libros/BasesConceptualesProg.pdf.

[MLBSO12] Pablo E. Martínez López, Eduardo Bonelli, and Federico Sawady O'Connor. El nombre verdadero de la programación. Una concepción de enseñanza de la programación para la sociedad de la información. *Anales del SSI 2012 - Simposio sobre la Sociedad de la Información, dentro de las 41ras Jornadas Argentinas de Informática (JAIIO)*, pages 1–23, 2012.
URL: http://41jaiio.sadio.org.ar/sites/default/files/1_SSI_2012.pdf.

[MLCAP17] Pablo E. Martínez López, Daniel A. Ciolek, Gabriela B. Arévalo, and Denise Pari. The Gobstones method for teaching computer programming. In *Proceedings of the 43rd Latin American Computing Conference - XXV Simposio de Educación Superior en Computación (SIESC'17), dentro de la XLIII Conferencia Latinoamericana de Informática (CLEI'17)*, CLEI. IEEE, 2017. ISBN 978-1-5386-3057-0,
URL: https://ieeexplore.ieee.org/document/8226428

[Pat81] Richard E. Pattis. *Karel the Robot: A Gentle Introduction to the Art of Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1981.

[SP88] P.C. Scholl and J.P. Peyrin. *Schémas algorithmiques fondamentaux: séquences et itération.* Université Joseph Fourier Institut Grenoblois d'études informatiques, 1988.

Formación docente y Currículum en el campo de las Ciencias de la Computación en dos provincias Argentinas

María Cecilia Martínez[1], Natalia Monjelat[2]

Palabras Clave: formación docente, currículum, ciencias de la computación.

1. Introducción

Diversos estudios señalan que la gran mayoría de los estudiantes secundarios tienen altas competencias de manejo de la información con el uso de tecnología, pero escasas competencias para crear, transformar, compartir información y entender el funcionamiento y uso de una computadora (Frailon, 2014). Tampoco dominan saberes que permitan transformar información a partir de algoritmos para automatizar el procesamiento de datos, aspectos esenciales de las Ciencias de la Computación (CC) (Denning, 1989).

En este contexto, en los últimos años diferentes países han realizado esfuerzos para incluir en la oferta educativa de primaria y secundaria contenidos de CC y más específicamente de Programación, ya que ésta permite acceder a diferentes dominios de la computación (Denning, 1989). En esta línea, el análisis del modelo israelí reveló que para lograr la incorporación de las CC en la escuela, se requeriría de 4 grandes elementos relacionados entre sí: 1) Currículum, 2) Investigación en el área de Enseñanza y Aprendizaje de la Computación, 2) Programas de formación docente y 4) Regulación sobre los títulos que habilitan la enseñanza de la computación en las escuelas (Hazza, Gael-Ezze y Blum, 2008). En Argentina, dos provincias con gran densidad poblacional (Córdoba y Santa Fe) fueron las primeras en el país en adherir a un Programa de la Fundación Sadosky, (presidida por el Ex Ministerio de Ciencia y Técnica de la Nación) para ofrecer Postítulos docentes en Enseñanza de las CC, de dos años de duración y una carga horaria de 600hs.
En esta ponencia mostramos cómo la elaboración de dichos Postítulos entre los años 2015 y 2017, supone un quiebre en el currículum, y prepara el camino para la inclusión de las CC en el contexto educativo argentino. Para ello, se analizarán los antecedentes en el currículo y la formación docente en torno a la enseñanza de las CC en las escuelas en estas provincias, identificando algunos criterios que guiaron el desarrollo de los contenidos, particularmente en cuanto a la Programación.

1 Universidad Nacional de Córdoba, cecimart@gmail.com
2 IRICE - CONICET-UNR, monjelat@irice-conicet.gov.ar

2. Aportes teóricos en torno a la noción de Currículum

El currículum es siempre un recorte de un campo disciplinar o área del conocimiento para ser enseñado, que luego las escuelas y los docentes van reconfigurando y adaptando, por lo que la neutralidad curricular no es posible (Terigi, 1999). Gvirtz y Palamidessi (1998) apuntan a tres ámbitos principales que influyen en la construcción del currículum: el campo cultural, el Estado y el mercado. En el caso de la informática podemos trazar una fuerte impronta del mercado que delinea los contenidos que deben incorporarse. En efecto, parte de la demanda de incluir computación en la escuela obligatoria proviene del sector de la industria del software que necesita más programadores (Patterson, 2006). En cuanto al Estado observamos que en Argentina hubo en la década pasada una fuerte inversión en ciencia y una apuesta a desarrollar la ciencia en nuestro país.

Esto requería de especialistas en el área de las CC puesto que el desarrollo digital atraviesa y potencia diferentes áreas del conocimiento. No obstante, las Universidades tienen el problema de que son pocos los estudiantes que eligen carreras relacionadas con la computación. En ese sentido, la introducción temprana de las CC podrían mejorar el desempeño de los estudiantes en estas áreas y generar más interés para considerar la computación como carrera. Desde el campo cultural, las brechas digitales (Benítez Larghi, 2011) y el bajo rendimiento de los estudiantes en habilidades relacionadas con la Alfabetización Digital, también demandan una democratización del saber informático. La escuela es la institución estatal que tradicionalmente se encarga de distribuir el acervo cultural. Estas múltiples demandas, han contribuido a que en algunas provincias se hayan elaborado orientaciones curriculares específicas en el campo de las tecnologías.

3. Métodos

Para analizar los antecedentes de las dos provincias en la promoción de contenidos de computación se recuperaron documentos que ofrecieran información sobre contenidos escolares de computación, y datos estadísticos sobre la oferta educativa en computación en cada provincia. Asimismo se recuperaron registros de las reuniones de los grupos de trabajo en la elaboración de los postítulos, que permitieron reconstruir los debates en torno a los contenidos y propuestas de formación docente así como las tensiones que se presentan en un programa de este tipo. Para identificar los principales contenidos y orientaciones de los diseños curriculares se elaboraron resúmenes y tablas que permitieron comparar y contrastar las propuestas entre provincias y entre niveles educativos. Se analizaron los temas emergentes de los registros de las reuniones.

4. Resultados

4.1. Antecedentes en las propuestas de formación docente en contenidos de las CC

Al momento de desarrollar el postítulo en la Provincia de Córdoba había escasa oferta de formación docente en Computación y nula en Programación. A nivel Universitario existe el Profesorado en CC de la Universidad Nacional de Río Cuarto. Además 5 Institutos de formación docente estatales dictan el profesorado en Educación Tecnológica. No obstante, el Plan de estudios no incluye contenidos específicos de Programación ni CC, sino que forma al futuro docente para ofrecer una enseñanza más general de la tecnología y los procesos tecnológicos. Los profesorados en educación inicial y primaria tampoco incluyen contenidos específicos de Computación, pero sí de lenguajes digitales y TIC (Ministerio de Educación de la Provincia de Córdoba).

Como resultado de la baja oferta formativa identificamos al menos que ⅓ de los docentes que enseñan materias relacionadas con la informática o la Programación no han tenido formación específica en Programación. Esta información se pudo reconstruir a partir de analizar las formaciones previas de los docentes que se inscribieron a nuestro postítulo. De los 644 postulantes al postítulo, 30,5% de esos docentes se desempeñan en materias relacionadas con la informática y sólo 30% tienen títulos habilitantes para enseñar Programación. No obstante, muchos de ellos no han tenido Programación en los planes de estudio de la carrera que cursaron. Por ejemplo, 40 pre inscriptos que enseñan Programación tienen el título de Profesores de Educación Tecnológica y otros 40 son Licenciados en Tecnología Educativa. 14% de nuestros aspirantes efectivamente enseñan materias que incluyen contenidos de Programación según consigna en sus perfiles, pero no pueden acceder al postítulo por no tener trayecto pedagógico previo que es un requisito según normativas nacionales. La gran mayoría de los pre inscriptos se desempeña en el secundario.

La Provincia de Córdoba acepta 14 títulos habilitantes y 18 títulos supletorios para poder acceder a un cargo de profesor de materias relacionadas con la Computación. Entre estos títulos se encuentra por ejemplo el de Técnico Superior de Instalación de Equipamiento Informático de Nivel Medio y de Administrador de Empresas Informáticas según consta de la lista de títulos habilitantes y supletorios de la Provincia.

Al momento de desarrollar el postítulo en Santa Fe, solo en 2 Institutos de formación docente se dictaba el profesorado en Educación Tecnológica, con un plan de estudios que no aborda contenidos específicos de CC y Programación. Por su parte, los profesorados en educación inicial y primaria cuentan dentro del campo de la formación general con un espacio curricular anual en tercer año denominado “Tecnologías de la Información y la Comunicación” donde se abordan las TIC desde la reflexión crítica sobre su impacto en el escenario socio cultural actual, reconociendo la existencia de una multiplicidad de lenguajes con los que se puede pensar, comunicarse y otorgarle sentido al mundo, identificando a su vez las posibilidades de uso pedagógico (Gobierno de Santa Fe, 2009a y 2009b).

Cuestionarios administrados a los docentes de primaria participantes de la primera cohorte del postítulo en didáctica de las CC dictado en Rosario (n=80), revelaron que un 68% ha participado de trayectos formativos en temáticas vinculadas a la educación y las tecnologías.

En este sentido, un 44% refiere a cursos cortos con duración y programas variados. A su vez un 19% señala haber participado del programa provincial “Tramas digitales” (creado en 2013) orientado a la integración de las nuevas tecnologías en todos los niveles y modalidades del sistema de educación provincial. Un 10% señala haber participado de postítulos nacionales en TIC para primaria o educación especial y otro 10% refiere a otras especializaciones o postítulos dictados por diferentes instituciones tanto privadas como públicas. Por otro lado, en cuanto a saberes específicos de las CC, un 21% tiene experiencia en Programación y dice conocer algunos de los programas más tradicionales que se utilizan para la enseñanza de estas temáticas en contextos escolares.

En síntesis, en ambas provincias predomina un enfoque de formación docente en TIC desde una perspectiva socio-cultural para incluir significativamente las tecnologías en las aulas sin que las CC aparezcan dentro de los trayectos de formación docente.

4.2. La inclusión de los contenidos de las CC en las bases curriculares

En el desarrollo de los planes de estudio, se procedió a identificar cuáles eran los contenidos relacionados con la computación que deberían transmitir los docentes en sus aulas. Se realizó entonces un análisis de las Bases Curriculares de las Provincias para el nivel inicial, primario y secundario. Cabe aclarar que en la provincia de Santa Fe no se dispone de una Ley Provincial de Educación Santafesina, por lo cual se rige por la Ley de Educación Nacional N° 26206/06 (Larraburu et al., 2016). En este sentido, los diseños jurisdiccionales disponibles para nivel inicial y primario se corresponden con las directrices de la Ley Federal de Educación Nº 24.195, sancionada en 1993. Estos diseños se encuentran en proceso de revisión, lo que hace que en la práctica, las instituciones educativas santafesinas del nivel inicial y primario, actualmente tomen como referencia los Núcleos de Aprendizaje Prioritarios (NAP) establecidos por el Consejo Federal de Educación. En estos documentos, se observan algunas referencias a contenidos tecnológicos para el nivel inicial en relación con el (Ministerio de Educación, 2011), mientras que para el nivel primario, se identifica un espacio curricular denominado “Educación tecnológica” desde 1° hasta 6° año (Ministerio de Educación, 2012a; Ministerio de Educación, 2012b).

Contrastando estos datos con los obtenidos en el contexto cordobés los resultados mostraron que los contenidos oficiales no abordaban la computación como objeto de estudio. Se observa entonces que desde el nivel inicial se propone estudiar la "tecnología" en su definición más amplia en tanto productos y técnicas elaboradas por el hombre. Se plantea el uso y reflexión sobre las TIC para gestionar información mas no para transformar datos, enfatizando un uso instrumental de la informática. Solo en el ciclo secundario con orientación en informática aparecen contenidos específicos de las CC (Resolución N° 2630/14; Ministerio de Educación de la Provincia de Córdoba, 2012).
Atendiendo a estas cuestiones, se expresó oportunamente a los referentes ministeriales de ambas provincias la ausencia de contenidos específicos en el currículum oficial, quienes

sugirieron que sean los contenidos del postítulo los que propongan las innovaciones curriculares y otorgaron libertad para la selección de los mismos.

4.3. El sentido de incluir la Programación en la escuela obligatoria y sus derivas en la selección de áreas de conocimiento dentro del campo

Además de identificar los contenidos de las CC que serían deseables para la escuela obligatoria los equipos de trabajo reflexionaron sobre el sentido de incluir estos contenidos en la escuela, considerando los perfiles de los docentes destinatarios y las características de los niveles educativos.

En Córdoba, se observó que el perfil formativo de la orientación en informática y la tecnicatura en informática del nivel secundario enfatizaban una formación para apoyar a los usuarios de software. En cambio, la tecnicatura en Programación proponía una formación para intervenir en las diferentes etapas del desarrollo de software. Para los niveles primarios y de educación general básica se propuso una formación para la comprensión. Acordamos que ofrecer contenidos de CC con énfasis en Programación era necesaria para la construcción de una ciudadanía alfabetizada en los lenguajes de nuestro tiempo. Aprender las bases de la Programación permitiría a los jóvenes comprender las lógicas de funcionamiento de la tecnología digital y desarrollar un pensamiento que incluya la posibilidad de cómputo en la resolución de problemas sociales y naturales. Teniendo en cuenta esta orientación, se recurrió entonces a analizar bases curriculares de Programación nacionales e internacionales. Se tomaron como ejes las propuestas de contenidos sugeridos por la Fundación Sadosky para la escuela obligatoria, los estándares de CUCEN (Consejo Universitario de Ciencias Exactas y Naturales) para el profesorado universitario en Informática y los marcos curriculares de la ACM (Association of Computing Machinery, 2016). Todos estos documentos que presentaban una selección curricular coincidían en dos grandes áreas:

-*Área de Programación* que incluía a) Fundamentos, Lenguajes Formales, Complejidad, Computabilidad, Autómatas b) Lenguajes de Programación, Resolución de problemas, Paradigmas (imperativo, lógico, funcional, objetos), Análisis comparativo de Lenguajes. c) Estructuras de Datos y Algoritmos d) Bases de Datos
-*Área de Sistemas Operativos* que incluía a) Arquitecturas, Sistemas Operativos y Redes b) Ingeniería de Software c) Interacción Máquina Hombre.
Además el contenido propuesto por la Fundación Sadosky incluía un Área de Didáctica y Pedagogía y de Representación de la Información que fueron agregados.

En Santa Fe, considerando la disposición del Consejo Federal de Educación que señaló al aprendizaje de la Programación de importancia estratégica para el Sistema Educativo Nacional durante la escolaridad obligatoria (Consejo Federal de Educación, 2015), acordamos que el postítulo tuviera como destinatarios a docentes en servicio del nivel primario. Asimismo, tomando como referencia los planes curriculares de otras instituciones para este nivel así como la experiencia de Córdoba, se decidió que los ejes fueran el

Pensamiento Computacional y la Programación. Con ello se buscaba que los maestros puedan apropiarse de esta forma de pensamiento y del potencial de la Programación para incluirlas en prácticas docentes situadas, interdisciplinares y no excluyentes, en línea a su vez, con la propuesta provincial de los NIC (Núcleos interdisciplinarios de contenidos) donde se plantean proyectos basados en diferentes problemáticas (Ministerio de Educación Provincia de Santa Fe, 2016)

En el caso de Córdoba, cuya experiencia fue crucial para el proceso rosarino, la selección de los contenidos definitivos se realizó a través de un ejercicio de Mapeo Invertido (Elmore, 1979) o Mapeo curricular. Después de comprender cuál era la selección de contenidos mínimos que otros organismos nacionales e internacionales proponían para la escuela primaria y secundaria definimos como equipo cuáles eran los objetivos de aprendizaje que buscábamos para los docentes: ¿Qué queremos que los docentes sepan y hagan al finalizar el postítulo? Después de profundos debates sobre el rol docente y los tipos de saberes que un docente debería dominar para diseñar experiencias de aprendizaje significativas, los objetivos generales decantaron en 3: apropiación de conceptos fundamentales de las CC que les permita seguir aprendiendo y comprendiendo los cambios tecnológicos, reconocimiento de las CC como disciplina con contenidos específicos que se diferencia con el uso de las TIC para poder abordar pertinentemente la enseñanza de la Programación y diseño de experiencias de enseñanza donde se ponga en juego el pensamiento computacional.

En el caso del postítulo de Santa Fe, nuestro objetivo general fue formar docentes capaces de experimentar y reflexionar críticamente acerca de los procesos de desarrollo del pensamiento computacional y la Programación, a los fines de construir las competencias adecuadas al nivel primario que posibiliten una práctica educativa innovadora con énfasis en la resolución de problemas mediante la producción colaborativa e interdisciplinaria de Tecnologías para la Inclusión Social. De esta forma, se entiende que tanto los procesos implicados en las prácticas docentes que incluyan contenidos relacionados con las CC, como los productos generados (programas, videojuegos, simulaciones, historias interactivas, etc.) puedan abordar problemáticas socioeducativas y regionales desde su complejidad (Monjelat, 2017), promoviendo un uso crítico de las herramientas y de conceptos en contextos proyectuales, a partir de poner en obra perspectivas pedagógicas activas, críticas e interdisciplinares (Casali et al., 2018a).

Estos objetivos guiaron el diseño curricular, proceso que implicó múltiples instancias de diálogo, debates y negociaciones entre los miembros de los equipos de trabajo, compuestos por especialistas de las ciencias de la educación y de las ciencias de la computación. En el caso cordobés, una tensión importante que emergió en los debates en torno al contenido, era sobre la profundidad con que se lo abordaría, ya que desde un esquema curricular más tradicional, se sostenía que la apropiación debía ser en profundidad. Recuperamos para estos debates los lineamientos internacionales (ACM, 2013) y aportes revisionistas (Krathwohl, 2002; Churches, 2008) que plantean una propuesta espiralada de apropiación conceptual, proponiendo entonces tres niveles de profundización en los contenidos:

- Familiaridad: los estudiantes adquieren nociones y significados centrales de los conceptos, pero no se requiere que los transfieran, apliquen o construyan nuevas ideas a partir de ellos.
- Uso: El estudiante es capaz de usar o aplicar el concepto en una forma concreta. Usar el concepto puede incluir, por ejemplo, utilizar apropiadamente el concepto en un programa, utilizar una técnica de prueba, o realizar un análisis determinado.
- Dominio: El estudiante es capaz de considerar un concepto desde múltiples puntos de vista y justificar la elección de un determinado enfoque para resolver el problema. Además, el estudiante puede producir nuevos saberes, productos o ideas a partir de relacionar conceptos.

Esta construcción espiralada del conocimiento, también fue retomada por la propuesta santafesina, tanto desde los contenidos presentes en los diferentes módulos como desde las propuestas de actividades y trabajos prácticos a lo largo de la especialización (Casali et al., 2018b).

5. Conclusiones

Identificamos que el desarrollo curricular para la escuela obligatoria en el área de tecnología ha ido evolucionando y ganando espacio a lo largo del tiempo. La enseñanza de la Programación implica una continuidad en los esfuerzos de alfabetización digital, pero a su vez una ruptura en el paradigma de inclusión de la tecnología en la escuela. Observamos que algunos avances en el área de Enseñanza y Aprendizaje de la CC materializados en selección de contenidos para propuestas curriculares, tensionan las propuestas de contenidos que habían elaborado las Provincias. En ese sentido la elaboración de los contenidos curriculares del postítulo producen un quiebre con los contenidos que venían ofreciendo las provincias, pero al mismo tiempo el postítulo abre un camino para implementar los nuevos NAP (Núcleos de Aprendizaje Prioritarios) aprobados a fines de 2018 que incluyen contenidos de Robótica y Programación. Los postítulos se convierten así en una herramienta de la política educativa para promover cambios en los contenidos.

6. Bibliografía

ACM (2013) Computer Science Curriculum.

ACM (2016). K-12 Computer Science Framework.

Benítez Larghi, S. et al. "De brechas, pobrezas y apropiaciones. Juventud, sectores populares y TIC en la Argentina." *revista Versión, México* (2011).

Casali, A., Zanarini, D., Monjelat, N., & San Martín, P. (2018a). Teaching and Learning Computer Science for Primary School Teachers: an Argentine Experience. In Proceedings of LACLO 2018 (pp. 1–8).

Casali, A., Zanarini, D., San Martín, P. S., & Monjelat, N. (2018b). Pensamiento Computacional y Programación en la Formación de Docentes del Nivel Primario. In G. Dapozo (Ed.), XX Workshop de Investigadores en Ciencias de la Computación (pp.

451–455). Corrientes, Argentina: Universidad Nacional del Nordeste. Facultad de Ciencias Exactas.

Churches, A. (2008) "Bloom's taxonomy blooms digitally." *Tech & Learning* 1: 1-6.

Consejo Federal de Educación (12 de agosto de 2015). Resolución CFE Nº 263/15. Buenos Aires, Argentina

CUCEN. Consejo Universitario de Ciencias Exactas y Naturales. Propuestas de Estándares para la formación del docente en informática.

Denning, P. J., Comer, D. E., Gries, D., Mulder, M. C., Tucker, A., Turner, A. J., & Young, P. R. (1989). Computing as a discipline. *Computer*, *22*(2), 63-70.

Elmore, R. F. (1979). "Backward mapping: Implementation research and policy decisions." *Political science quarterly* 94 (4)., pp. 601-616.

Fraillon, J., Ainley, J., Schulz, W., Friedman, T., & Gebhardt, E. (2014). Preparing for life in a digital age: The IEA International Computer and Information Literacy Study international report. Disponible en https://research.acer.edu.au/cgi/viewcontent.cgi?article=1009&context=ict_literacy

Gvirtz, S. y Palamidessi, M. (1998) *El ABC de la tarea docente: currículum y enseñanza*. Vol. 1. Buenos Aires: Aique.

Gobierno de Santa Fe, Ministerio de Educación. (2009a). Profesorado de Educación Primaria. Diseño Curricular para la Formación Docente. Santa Fe, Argentina.

Gobierno de Santa Fe, Ministerio de Educación. (2009b). Profesorado de Educación Inicial. Diseño para la Formación Docente. Santa Fe, Argentina.

Hazzan, O., Gal-Ezer, J., & Blum, L. (2008). A model for high school computer science education: The four key elements that make it!. *ACM SIGCSE Bulletin*, *40*(1), 281-285.

Krathwohl, D. R. "A revision of Bloom's taxonomy: An overview." *Theory into practice* 41.4 (2002): 212-218.

Larraburu, S. R., Chapani, D. T., Garcia Romano, L., & Farré, A. (2016). Las políticas públicas y la implementación de las TIC en las clases de Ciencias Naturales. Aula Universitaria, (18), 20–28. https://doi.org/10.14409/au.v0i18.6554

Ministerio de Educación Provincia de Santa Fe (2016). Núcleos Interdisciplinarios de Contenidos: la educación en acontecimientos. Documento de desarrollo curricular para la educación primaria y secundaria. Santa Fe, Argentina.

Ministerio de Educación. (2011). Núcleos de Aprendizajes Prioritarios. Educación Inicial. Buenos Aires, Argentina.

Ministerio de Educación. (2012a). Núcleos de Aprendizajes Prioritarios. 1o ciclo educación primaria 1°, 2° y 3° Años. Buenos Aires, Argentina.

Ministerio de Educación. (2012b). Núcleos de Aprendizajes Prioritarios. 2o ciclo educación primaria 4°, 5° y 6° Años. Buenos Aires, Argentina.

Ministerio de Educación de la Provincia de Córdoba (2012) Diseño Curricular de Educación Secundaria.

Monjelat, N. (2017). Programming technologies for social Inclusion: An experience in professional development with elementary teachers. In 12th Latin American Conference on Learning Objects and Technologies, LACLO 2017 (Vol. 2017–January). https://doi.org/10.1109/LACLO.2017.8120901

Patterson, D. A. "Computer science education in the 21 st century." *Communications of the ACM* 49.3 (2006): 27-30.

Resolución N° 2630/14. Diseño curricular. Educación Secundaria Orientada Provincia de Santa Fe. Ministerio de Educación de Santa Fe. Santa Fe, 30 de diciembre de 2014.

Terigi, F. (1999). Conceptos para el análisis de políticas curriculares. En *Currículo: itinerario para aprehender un territorio.* Buenos Aires: Editorial Santillana, 115-136.

Enseñar a Enseñar Ciencias de la Computación. Una experiencia sobre políticas educativas y contenidos de Ciencias de la Computación

Claudia Queiruga[1], Claudia Banchoff Tzancoff[2], Soledad Gómez[3], Paula Venosa[4]

Resumen[5]: El trabajo que se presenta pretende hacer un análisis preliminar de los logros y las dificultades encontradas durante el dictado de los dos primeros módulos del postítulo "Especialización en didáctica de las Ciencias de la Computación" de la UNLP-ISFD Nº 95 de La Plata: "Marco político pedagógico" y "El algoritmo como objeto de aprendizaje y forma de organización del pensamiento".

Para el análisis se consideran dos tensiones propias de la práctica educativa: la tensión entre teoría y práctica, y las decisiones pedagógicas didácticas que sostienen el enfoque problematizador en ambas propuestas. Este trabajo se fundamenta en las percepciones de los equipos docentes de ambos módulos recabadas a partir de reuniones de trabajo en las que se presentaron de forma incipiente las principales inquietudes respecto de las respuestas y devoluciones recibidas en las clases por parte de los docentes cursantes. En este sentido, se considera que el enfoque pedagógico propuesto en la carrera y, la multiplicidad y variedad de perfiles docentes, contribuyen a que las percepciones en relación al sentido de la propuesta fuera diverso.

Se reconoce una valoración positiva, por parte de los docentes cursantes, del enfoque de contenidos específicos de Ciencias de la Computación, así lo evidencia la encuesta implementada, desde la cual se retomarán algunos datos. Sin embargo la comprensión de esos contenidos y su comparación con el enfoque centrado en TIC, trajo consigo una serie de situaciones que motivaron el desarrollo de propuestas didácticas que sirvieron para definir características específicas de ambos enfoques. A su vez, la convivencia de distintos perfiles de formación en las cursadas, docentes con mayor orientación técnica y otros de perfil más pedagógico, propició un rico espacio de debate respecto de los contenidos específicos y de las propuestas didácticas que se diseñaron para la enseñanza de los mismos. Estas experiencias y otras que incluyen los procesos de toma de decisión de los cuerpos docentes de ambos módulos, son objetivo de análisis y reflexión para este equipo de trabajo, sobre los cuales se sientan las nuevas propuestas del segundo ciclo del dictado del postítulo y sobre las cuales consideramos que debemos seguir trabajando en la enseñanza de las Ciencias de la Computación.

Palabras claves: Enseñanza, Programación, Formación docente, Didáctica, Escuelas

---

1 Laboratorio de Investigación en Nuevas Tecnologías Informáticas (LINTI), Facultad de Informática, Universidad Nacional de La Plata (UNLP) claudiaq@info.unlp.edu.ar

2 LINTI, Facultad de Informática – UNLP cbanchoff@info.unlp.edu.ar

3 LINTI, Facultad de Informática – UNLP sgomez@info.unlp.edu.ar

4 LINTI, Facultad de Informática – UNLP pvenosa @info.unlp.edu.ar

5 Sin desconocer la pauta sexista del idioma español ni las novedosas prácticas de lenguaje inclusivo, a los fines de facilitar la lectura, en este trabajo se usará el genérico masculino al referirnos a las categorías de personas.

Contexto

La sociedad actual, denominada por algunos autores Sociedad de la Información o Sociedad del Conocimiento y más precisamente Capitalismo Informacional por Manuel Castells, identifica la etapa del capitalismo en la que el conocimiento digital forma parte de todos los procesos productivos. Esta realidad plantea en el campo educativo la formación de ciudadanos que puedan comprender los lenguajes digitales, ubicándolos como sujetos críticos y creadores de innovaciones con tecnologías digitales y, que a su vez puedan beneficiarse de las oportunidades que brinda el software como motor de desarrollo económico y social. Los enfoques acerca de cómo introducir en las curriculas de los sistemas educativos obligatorios la Informática como campo del saber, es un tema de investigación actual que cuenta con una gran variedad de trabajos publicados en revistas científicas, de divulgación y paneles de expertos de todo el mundo que reflexionan en torno a este cuestión. Nuestro país no es ajeno a este debate y múltiples programas federales y normativas dan cuenta de ello, el proyecto “Program.AR” (Program.ar, sf), la resolución del Consejo Federal de Educación (CFE) del 2015, que declaró el aprendizaje de la programación como una herramienta de "...importancia estratégica para el sistema educativo argentino", que será enseñada durante el ciclo de escolaridad obligatoria en todas las escuelas de la Argentina, en 2016 el “Plan Nacional Integral de Educación Digital” (PLANIED, sf), que pone el acento en ampliar la formación tecnológica, educando a los jóvenes en la comprensión de cómo funcionan las tecnologías y más recientemente, el proyecto “Secundaria 2030” (Secundaria 2030, sf) aprobado por el CFE que propone incorporar gradualmente un enfoque de enseñanza basado en capacidades y competencias digitales transversales y, en 2018 la inclusión de contenidos de programación y robótica en los Núcleos de Aprendizaje Prioritarios (NAPs) de todo el territorio nacional.

En respuesta a atender la necesidad de formación de un campo caracterizado por las escasas referencias en relación a contenidos y metodologías de enseñanza, la Fundación Sadosky lanzó en 2016 una convocatoria a universidades nacionales con carreras de Informática, orientada a diseñar e implementar una especialización en didáctica de la Informática en asociación con Institutos Superiores de Formación Docente (ISFD) de sus jurisdicciones. La UNLP en asociación con el ISFD Nº95 de La Plata elaboró el diseño curricular de la “Especialización docente en didáctica de las Ciencias de la Computación” dirigido a docentes de nivel secundario con formación de base en la disciplina o afines como matemática, física, química y tecnología (Queiruga C., Banchoff Tzancoff C., Venosa P, Gómez S., Morandi G., 2019). En 2018 se comenzó a dictar la carrera, habiendo concluido, a la fecha, seis (6) de los 8 módulos específicos y la Práctica Situada I.

En este trabajo se presenta un análisis del dictado de los dos primeros módulos: “Marco Político Pedagógico” y “El algoritmo como objeto de aprendizaje y forma de organización del pensamiento”.

En el módulo denominado “Marco Político Pedagógico” se trabajaron los aspectos que justifican, de algún modo el abordaje de esta temática en la escuela secundaria. Se

analizaron los marcos político-pedagógicos de la educación secundaria y, específicamente, el papel de la formación en Ciencias de la Computación en este marco, identificando las potencialidades y alcances de la enseñanza de las Ciencias de la Computación en el contexto socio-histórico actual atravesado por el desarrollo tecnológico y la innovación en este campo. El enfoque del pensamiento computacional y el análisis del concepto de soberanía tecnológica y sus relaciones con el desarrollo de software libre y software privativo, fueron temas abordados para problematizar los sentidos pedagógicos respecto de la enseñanza de la Ciencia de la Computación. Los contenidos mínimos del módulo "Marco Político Pedagógico" comprenden:

- La contextualización de la formación en Ciencias de la Computación y la diferenciación de otras formas de inclusión de tecnología en el aula tales como ofimática, tecnología educativa.
- Nuestra vida cotidiana organizada en torno a y por las computadoras y sistemas de computadoras: implicancias y consecuencias de vivir en un mundo de "sistemas inteligentes" de los que desconocemos el funcionamiento.
- Los contenidos que forman parte de la Ciencias de la Computación y las habilidades asociadas. El posicionamiento sobre el Pensamiento Computacional.
- Por qué enseñar Ciencia de la Computación en países en vías de desarrollo, la cuestión de la soberanía tecnológica versus la dependencia tecnológica. La tensión entre la formación de recursos humanos calificados y la demanda del mercado de mano de obra en el sector; su relación con la falta de presencia de Ciencias de la Computación en la escuela.

Los contenidos del "Marco Político Pedagógico" fueron abordados en su totalidad, siendo diferentes los tiempos y la profundización de los mismos, condicionados por las zonas de interés que los docentes cursantes demostraron en cada caso. Sobre este punto es importante destacar que hubo confusión, por parte de los docentes cursantes, respecto del enfoque pedagógico del postítulo, orientado a las Ciencia de la Computación y con especial atención en programación. Los docentes inscriptos, en su mayoría, poseen experiencias de trabajo áulico dentro de espacios curriculares con enfoque más generalista sobre las tecnologías, como lo es la materia NTICx del diseño curricular de educación secundaria de la provincia de Buenos Aires. Este diagnóstico general, respecto de los perfiles docentes, implicó la primera toma de decisiones sobre el contenido, y la primera pregunta pedagógica para el equipo docentes a cargo de la propuesta, ¿qué queremos transmitir sobre el campo de la Ciencia de la Computación como propuesta de enseñanza?.

En el módulo se desarrollaron cuatro (4) núcleos temáticos, que tuvieron como objetivo problematizar el emergente campo de la enseñanza de la Ciencia de la Computación en la escuela secundaria. Dentro del primer eje se recupera la especificidad del enfoque propuesto en la carrera, y se desarrolla su diferenciación con el enfoque TIC. Se establece una articulación con el desarrollo de un tipo de pensamiento, abordando el pensamiento computacional, sus principales definiciones y enfoques de trabajo en el eje 3. Y se problematiza desde un posicionamiento pedagógico crítico la formación en estas temáticas,

dentro del eje 4, abordando la posibilidad de una soberanía tecnológica, en relación con los sujetos de la educación y sus procesos de producción de conocimiento. El eje 2 propone pensar a las tecnologías desde un escenario cotidiano, naturalizadas dentro de las prácticas más elementales. Para el desarrollo de las clases, este problema se recuperó de forma transversal para abordar cada uno de los temas propuestos.

En base al primer interrogante "¿qué enseñar?", se priorizaron para el desarrollo los ejes 1, 3 y 4, porque se los consideró esenciales para la presentación del campo, sus problemáticas y desafíos para la enseñanza, a su vez porque el eje 2 es conocido por los docentes, lo cual, facilitó la integración con los ejes 1, 3 y 4 a partir de pensar las tecnologías digitales en nuestros ámbitos cotidianos.

El equipo docente que trabajó en este módulo estuvo formado por tres (3) profesoras y dos (2) ayudantes, con una formación en Comunicación y Ciencias de la Educación.

El módulo denominado "El algoritmo como objeto de aprendizaje y forma de organización del pensamiento" se centró fundamentalmente en analizar y proponer estrategias de enseñanza dirigidas a la formulación de soluciones a distintos problemas a través de la construcción de algoritmos. Se trabajaron los aspectos básicos de la programación como ser los conceptos de "programa como secuencia de instrucciones", una introducción a la lógica proposicional, estructuras de control y la noción de procedimientos para abstraer comportamientos y modularizar las soluciones. En este módulo se trabajó con actividades desconectadas, principalmente lúdicas. Los contenidos mínimos del módulo "El algoritmo como objeto de aprendizaje y forma de organización del pensamiento" comprenden:

- La enseñanza y el aprendizaje del concepto de algoritmo como base para la formación en programación. El proceso de abstracción en la resolución de problemas.
- La recuperación de estrategias lúdicas basadas en diversos tipos de juegos popularmente conocidos y de circulación cotidiana, tales como juegos de ingenio, juegos de naipes, de tablero, trucos de magia, etc para la apropiación del pensamiento algorítmico.
- La utilización de algoritmos que resuelven los problemas de computación típicos, ejemplo de ello son algoritmos de ordenación y de búsqueda y uso de elementos concretos para su materialización.
- La utilización, como estrategia didáctica, la resolución de problemas relativamente complejos en equipos de trabajo, usando la técnica de descomponer el problema en partes simples y manejables en la que cada miembro del equipo pueda aportar sus ideas, experiencias y habilidades.

El equipo docente que trabajó en este módulo estuvo formado por tres (3) profesoras y cuatro (4) ayudantes, con una formación en Informática. La razón por la cual en este módulo se contó con un mayor número de ayudantes es que la mayoría de los temas abordados se basaron en actividades prácticas sobre la resolución de problemas utilizando el enfoque

algorítmico y dado el perfil heterogéneo de formación del alumnado respecto de los contenidos específicos de Informática fue necesario incrementar la relación docente-alumno.

El análisis de estas experiencias se transforma en objetivo de este trabajo, en tanto permite reflexionar críticamente sobre una práctica docente específica. Y esa reflexión nos posibilita construir una contribución a futuro en cada uno de los módulos a dictarse dentro de la carrera.

Marco conceptual

El desafío de planificar el desarrollo de una propuesta didáctica que represente el enfoque crítico problematizador que atraviesa esta carrera de especialización, no fue sencillo. En opinión de Edelstein y Litwin (1993 : 79-86), la enseñanza es artesanal al estar *"...teñida de componentes éticos, morales, políticos y normativos"*. Y *"...debido a su inevitable componente moral, la actividad educativa está recorrida por el carácter cuestionable y problematizador de cada una de sus decisiones"*. En este sentido, el proceso de toma de decisión de ambos equipos docentes, estuvo condicionado por dos tensiones centrales: la tensión entre teoría y práctica, presente en cada práctica educativa y la tensión entre didáctica y pedagogía.

Estas tensiones estuvieron condicionadas por el contexto, dado que nos encontramos frente una carrera que se propone no sólo enseñar contenidos del campo de la Ciencia de la Computación, sino que trabaja sobre las formas y los sentidos en que ese campo de conocimientos sea transmitido, la propuesta de la carrera propone enseñar a enseñar contenidos de un campo nuevo y emergente en la escuela. Aquí reconocimos la tensión entre teoría y práctica, dado que debíamos en primera instancia definir esa relación, si basamos nuestras propuestas poniendo el énfasis en la teoría priorizando la enseñanza de los contenidos o focalizamos en los procesos de aprendizaje inclinándonos más hacia la práctica.

Ninguno de los extremos lograba encarnar nuestras propuestas tal y cual las habíamos pensado, en relación al objetivo de la carrera, metodológicamente adherimos a la idea de centrar nuestras clases en el desarrollo práctico para llegar a los contenidos y de alguna manera eso nos permitiría articular ambos polos. Nuestras intervenciones estuvieron planificadas desde *"la Didáctica Crítica, donde el aprendizaje es concebido como un proceso que manifiesta constantes momentos de ruptura y reconstrucción, las situaciones de aprendizaje cobran una dimensión distinta a los planteamientos mecanicistas del aprendizaje, pues el énfasis se centra más en el proceso que en el resultado"* (Morán, 1996: 194).

Por otro lado nos valimos de categorías del campo de la educación que nos permitieron dimensionar la complejidad de los procesos de formación que pretendíamos lograr, a la vez que nos habilitaba la posibilidad de reflexionar sobre nuestra propia práctica docente, no sólo por las decisiones que debíamos tomar sino por el posicionamiento que nos

propusimos. Es nuestra intención promover procesos de formación que les permita a los docentes cursantes la implementación de contenidos del campo de la Ciencia de la Computación en espacios curriculares que no corresponden a estos contenidos específicos, dado que consideramos que la inexistencia de materias específicas en el diseño curricular actual de la educación secundaria de la provincia de Buenos Aires, no puede seguir siendo una limitante para el abordaje de estos contenidos. Aquí entonces aparece la transversalidad y la necesidad de interacción con otros campos disciplinares, ¿cómo promovemos procesos que involucren contenidos de Ciencias de la Computación para otras materias? ¿cómo enseñamos a programar para enseñar no solo programación sino contenido de prácticas del lenguaje, por ejemplo? En este ejercicio de fundamentación de un campo emergente en la escuela secundaria y de una didáctica en construcción, fue que decidimos articular las producciones finales de ambos módulos, promoviendo el desarrollo de una propuesta de enseñanza que trabaje con los contenidos abordados. Desde el módulo "Marco Político Pedagógico" se solicitó a los docentes que utilicen los marcos normativos vigentes para diseñar, fundamentar y objetivar una propuesta de enseñanza que recupere contenidos del campo de la Ciencia de la Computación en su materia. A su vez dentro del módulo "El algoritmo como objeto de aprendizaje y forma de organización del pensamiento", se solicitó a los docentes cursantes la elaboración de una secuencia didáctica para la enseñanza de algoritmos que tomase como campo de aplicación el de su materia. Fue necesario un trabajo previo de selección de los contenidos de las asignaturas y cómo relacionarlos con los conceptos de algoritmos trabajados en el módulo. A modo de ejemplos, una profesora de Biología tomó como tema "El ADN nuclear como molécula portadora de información. Estructura molecular del ADN" y a partir de allí elaboró una secuencia didáctica, en la que se trabajaron los conceptos de *autómatas y abstracción,* en una dinámica de grupos, cuya actividad consistió en realizar un algoritmo que construya una cadena de ADN; una profesora de arte que encuadró su trabajo en el espacio curricular de Plástica Visual y cuyo tema a trabajar fue el arte digital a través de los hiloramas. La secuencia didáctica propuesta relaciona este tema con el concepto de *abstracción reconociendo los patrones* que forman los hiloramas y luego define el algoritmo que lo podría generar. La integración de las propuestas de ambos módulos nos permitió fortalecer el trabajo de posicionamiento y enfoque para el desarrollo de una propuesta áulica concreta, que los docentes trabajaron situándose en sus propios escenarios institucionales.

Este ejercicio nos llevó directamente al campo de la didáctica, por acaso como lo explica Zambrano (2015) en relación al fundamento de este campo *" .. el tema clave de su objeto se expresa en las preguntas ¿cómo aprenden los niños? ¿cómo enseñar el contenido —conceptos, nociones, elementos— de una disciplina? y ¿cómo generar una adecuada apropiación del contenido de las disciplinas en términos de saber? Si esto sucedía del lado del aprendiz, del lado del profesor la pregunta clave seguirá siendo ¿Qué enseña un profesor y cuál es su relación con el conocimiento de su disciplina?"*. Frente a nuestras dificultades didácticas, las tensiones que aparecieron en este proceso fueron comunes en los dos módulos que tomamos como referencia en este trabajo, y que son parte del primer ciclo de implementación de la carrera. Al ser los espacios de apertura, cargaban con la

responsabilidad de hacer claro el enfoque pedagógico y didáctico de la carrera. En este marco, fue necesario pensar intervenciones coherentes a lo planteado, que motiven el interés por los contenidos de Ciencias de la Computación desde una perspectiva de complejidad pero sobretodo de la enseñanza, como explican Martínez y Echeveste (2017) *"En un contexto donde acechan las lógicas individualistas y meritocráticas, donde el sujeto hace un uso aislado de las tecnologías, como docentes podemos 'resistir' enseñando mediante apuestas colectivas de trabajo comprometidas con el desarrollo de modos de pensamientos complejos y críticos"*.

Análisis de las experiencias

Los módulos "Marco Político Pedagógico" y "El algoritmo como objeto de aprendizaje y forma de organización del pensamiento", se dictaron en forma simultánea, en encuentros intercalados. Los encuentros se realizaron los días viernes por la tarde y sábados durante todo el día. Esta disposición de las clases fue pensada de manera de poder encarar algunas de las actividades en forma articulada entre ambos módulos. Luego del dictado de los mismos, esta planificación de clases fue modificada para los siguientes módulos, principalmente dado que se observó que algunas actividades en las que se trabajaron los conceptos de algoritmos, que en términos generales fueron las que presentaron mayores dificultades, quedaron demasiado espaciadas entre sí.

El análisis de los módulos dictados se puede organizar en torno a las siguientes categorías, a partir de las observaciones de los equipos docentes:

- Resistencia a una didáctica crítica (tensión con las formas de enseñanza tradicionales).
- Tensión entre los contenidos específicos de la disciplina y las propuestas transversales.
- Dificultad en reconocer el cuerpo de conocimiento del campo de la Ciencia de la Computación en el que se están formando.

Estas problemáticas se evidenciaron más acentuadamente en la etapa de producción de contenidos: varios docentes que se encuentran a cargo de cursos de Informática cuestionaron la didáctica planteada argumentando cierta desconfianza en la construcción de situaciones problema, dado que la manera tradicional en la que presentan los temas de programación en sus clases les da buenos resultados: en varias oportunidades se plantearon secuencias didácticas donde en una actividad inicial se da una explicación sobre el tema en cuestión, usando definiciones y luego se propone una serie de hacer ejercicios sobre el tema. Esta situación se evidenció en las propuestas didácticas que se les acercaron desde los módulos, donde el trabajo estuvo planificado desde un primer momento en promover un espacio de construcción. Las propuestas que acompañaron ambos módulos estuvieron pensadas para que los docentes descubran el contenido, aún en aquellos casos en que éstos les resultaban conocidos; se motivó a pensar desde un enfoque basado en la enseñanza. Esta motivación se llevó a cabo desde dos acciones concretas: seguimiento y evaluación en proceso. Con estas acciones se pretendió problematizar los modos en que el

contenido es puesto en juego a partir de prácticas de producción concreta, donde además de dar cuenta de los temas y del enfoque se invitó a los docentes a problematizar las propias prácticas áulicas. A modo de ejemplo para el eje de contextualización de la formación en Ciencia de la Computación, se propusieron varias dinámicas de reflexión grupal, donde los docentes debían: armar un campo de preguntas, sobre la implementación de contenidos de Ciencias de la Computación en la escuela que recupere las características específicas de este tipo de conocimiento, en relación con el marco normativo revisado. A su vez, los cursantes compartieron las reflexiones sobre las propias experiencias, con el objetivo de analizar las dificultades y desafíos comunes. En este ejercicio se pudieron rastrear los campos de preguntas que los docentes tienen sobre la incorporación de tecnologías y de contenidos de Ciencias de la Computación para trabajar las tensiones entre teoría y práctica por ejemplo, donde se reconocían los temas específicos de Ciencia de la Computación pero a la hora de pensar prácticas educativas, las propuestas recuperaban usos de tecnologías digitales. Este ejercicio, fue uno de los primeros, y el que nos dio la certeza de que era necesario profundizar el debate y la reflexión respecto del enfoque disciplinar y didáctico que tiene la carrera. Fue un interesante proceso de conceptualización que facilitó el abordaje y la línea de trabajo para el resto de los módulos, funcionando como módulo de acuerdos, en tanto definimos qué son las Ciencias de la Computación, el enfoque pedagógico crítico para su enseñanza y diseño de las estrategias didácticas.

Para ejemplificar las tensiones generadas con las formas de enseñanza tradicionales, podemos mencionar la manera en que se introdujo el uso de estructuras condicionales en el módulo dedicado a algoritmos. Se comenzó con un juego basado en el tradicional juego de "Adivina el personaje", que permitió abordar la noción de proposición lógica, valor de verdad, tablas de verdad y algoritmos en escenarios cambiantes. Ante esta propuesta, algunos docentes, fundamentalmente los que están a cargo de materias de programación, mayoritariamente provenientes de escuelas secundarias técnicas con orientación Informática, plantearon que les resultaba más útil y sencillo abordar directamente el tema a partir de la sintaxis de la sentencia condicional (if-then-else), realizar una explicación de cómo se usa dicha sentencia y luego solicitar a sus estudiantes que practiquen ejercicios en los que deben aplicar lo explicado, a modo de fijar el tema. Esta resistencia emerge frecuentemente y es tema de debate e intercambio a lo largo de los módulos.

Hubo casos concretos en los que fue complejo planificar propuestas para el contenido de Ciencias de la Computación, dado que la realidad es que los docentes se desempeñan en espacios que no son específicos de este campo y que muchas veces según los diseños curriculares vigentes hasta la transversalidad es un desafío. En estos casos el seguimiento, las instancias de devolución y reentrega posibilitaron un diálogo que permitió no solo pensar las estrategias sino la realidad de cada escuela.

Asimismo, varios docentes acordaron con el enfoque didáctico propuesto, sus producciones presentan claramente problemas contextualizados que recrean situaciones reales, sin embargo su objeto de aprendizaje se remite a otros campos del saber y la Ciencias de la Computación es un auxiliar. Esto último podría estar vinculado al uso eficiente

de herramientas informáticas, concepción bastante arraigada en las escuelas, a la falta de legitimación de la Ciencia de la Computación como campo del saber y a la formación inicial de varios de estos docentes que se corresponde con campos diversos (química, biología, letras, arte, etc). Esta situación se evidencia por la impronta de las políticas curriculares que definieron espacios en las escuelas desde una perspectiva TIC, donde el acento estuvo puesto en la incorporación y uso de tecnologías, promoviendo prácticas docentes que utilicen tecnologías digitales.

Para continuar la problematización de estos contenidos, que en estos módulos se dio un poco por la incorporación de contenidos específicos pero también por la apuesta de un enfoque que reivindica, no sólo la formación docente, sino la enseñanza, esta carrera definió dos espacios de prácticas situadas. Los mismos se desarrollan al finalizar cada uno de los años de la carrera y tienen como objetivo revisar las propuestas didácticas que se trabajan en los módulos de contenido específicos, para su implementación en las aulas propias. Entendemos a estos espacios como desafíos, en tanto la mayoría requieren de propuestas de transversalidad o de interacción con otros espacios dentro de las escuelas. En el nivel II de la Práctica Situada se profundiza esta instancia de diseño para pensar proyectos institucionales que aporten y beneficien la integración de contenidos de Ciencia de la Computación en las escuelas. Recuperar las realidades institucionales, la experiencia de los docentes dada su trayectoria y promover un seguimiento a lo largo de los módulos de la carrera resulta fundamental para que el resultado de la Práctica Situada II responda a los objetivos que nos propusimos en el diseño de la carrera.

A partir del análisis de las encuestas de percepción de los docentes sobre los contenidos y la didáctica, administradas al finalizar los primeros dos módulos, las Fig. 1 y Fig. 2 permiten observar que entre el 80% y 90% de los docentes cursantes evalúan positivamente **l**os contenidos de ambos módulos. Aquí es relevante tener en cuenta que varios de los cursantes son docentes de NTICx y de espacios curriculares de escuelas secundarias técnicas con orientación informática:

- En el módulo “El algoritmo como objeto de aprendizaje y forma de organización del pensamiento”, el 55% evalúa a los contenidos conocidos con enfoque novedoso, y el 36% los considera desconocidos pero asequibles. En este último dato aparece la diversidad de formaciones disciplinares de los cursantes.
- En el módulo “Marco Político Pedagógico”, el 50% evalúa a los contenidos conocidos con enfoque novedoso, un 15% desconocidos pero asequibles y un 30% como demasiados exhaustivos pero bien articulados. Estas últimas dos categorías podrían pensarse como una, en el sentido que son valorados como asequibles, independientemente de ser conocidos o no.

En ambas experiencias las percepciones respecto de los contenidos específicos arrojan altos niveles de interés y pertinencia, pero a la vez un buen índice de aceptabilidad. En términos de intereses, la predisposición de los docentes sobre los temas propuestos

también se analiza en términos favorables, teniendo en cuenta que en la cohorte hay un alto porcentaje de docentes de otras disciplinas.
En relación a la propuesta didáctica, como muestran las Fig. 1 y Fig. 2, sobre las intervenciones pedagógicas en ambos módulos, las encuestas arrojan que entre el 50% y 60% de los docentes cursantes valoraron como innovador el enfoque propuesto y las restantes respuestas se distribuyen de manera diferente:

- En el módulo "El algoritmo como objeto de aprendizaje y forma de organización del pensamiento" el 27.3% evalúa innovador el enfoque utilizado, favoreciendo la asimilación de los contenidos y el 21.2% como movilizador en términos de permitir pensar nuevas maneras de dar clases. El 18.2% indicó que le llevaba mucho tiempo resolver los problemas planteados. El 30% restante de las respuestas se distribuyen entre múltiples opciones que podrían atribuirse a la diversidad de disciplinas dado que en varias se evoca a ser profesor de otra área y no saber cómo dar sus clases de esta manera.
- En el módulo "Marco Político Pedagógico", el 32.5% de las respuestas obtenidas aseguran que les pareció innovador y les permitió asimilar de forma sencilla temas trabajados. A su vez el 30% aseguró que les permitió pensar nuevas maneras de dar sus clases y el 15% marcó que llevaba demasiado tiempo resolver cada uno de los problemas presentados.

En el análisis de estas respuestas es necesario reconocer que si bien ambos módulos tienen por objetivo fortalecer las prácticas de enseñanza sobre contenidos de Ciencia de la Computación, son muy diferentes los enfoque y, por ende, las estrategias didácticas implementadas. Mientras el "Marco Político Pedagógico" pretende dar fundamentos para la implementación de esos contenidos en las escuelas, el módulo de "El algoritmo como objeto de aprendizaje y forma de organización del pensamiento" promueve las bases disciplinares para el abordaje de un campo de conocimiento específico.

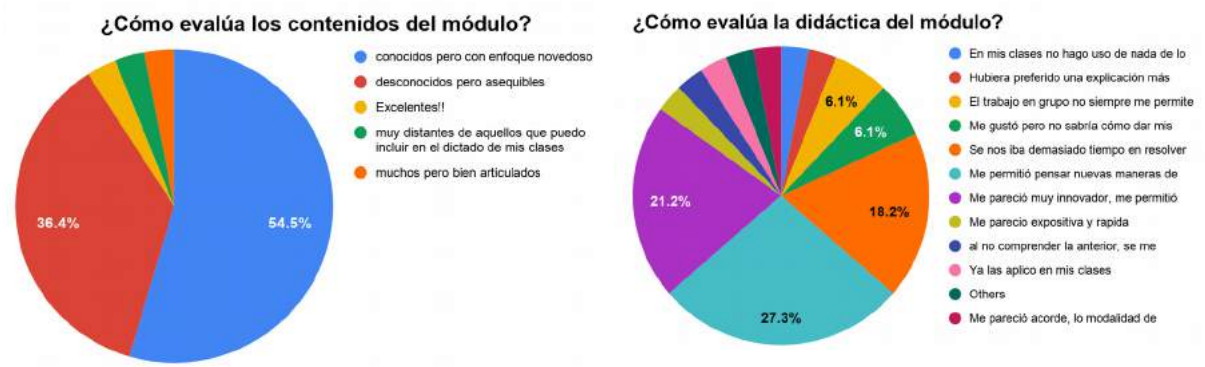


Figura 1. Módulo "El algoritmo como objeto de aprendizaje y forma de organización del pensamiento": percepción de los docentes cursantes sobre los contenidos y la didáctica.

Figura 2. Módulo “Marco Político Pedagógico”: percepción de los docentes cursantes sobre los contenidos y la didáctica.

Consideraciones para seguir avanzando

La realización de este trabajo nos permitió plasmar la experiencia realizada y volver sobre los procesos de reflexión y de toma de decisión que nos asaltaron en el proceso. Las situaciones enunciadas en este trabajo fueron tenidas en cuenta a la hora de plantear los siguientes módulos, por lo que se establecieron nuevos acuerdos poniendo especial atención en el campo de formación del postítulo. Los siguientes módulos han finalizado:

- La enseñanza de la programación a través de lenguajes visuales basados en bloques.
- Las computadoras y los sistemas operativos.
- Redes de datos e Internet.
- La enseñanza de la programación en lenguaje real de producción de software con especial atención al Software libre.

A su vez concluyó la primera etapa de la Práctica Profesional Situada.
Compartir la experiencia del proceso realizado, es para este equipo, necesario en tanto entendemos que a partir de las experiencias podemos construir un camino más sólido hacia la enseñanza de la Ciencia de la Computación en las escuelas. Carreras como este postítulo permitirán contar con la formación docentes necesaria, para espacios curriculares específicos dentro de las curriculas, y eso nos coloca un paso más cerca del objetivo.

Bibliografía

Edelstein, G. y Litwin, E. (1993). “Nuevos debates en las estrategias metodológicas del curriculum universitario”. En *Revista Argentina de Educación* Nº 19, Año XI. Buenos Aires.

Martínez, C., Echeveste, M. (2017). *Aprender a programar para integrar(nos)*, Serie Cuadernos para la enseñanza, Instituto de Capacitación e Investigación de los Educadores de Córdoba de la Unión de Educadores de la Provincia de Córdoba (ICIEC-UEPC). Disponible en: https://bit.ly/2n4QNbJ. Último acceso: 14/04/2019.

Morán, Oviedo Porfirio. (1996). “La instrumentación didáctica en la perspectiva de la didáctica crítica” en *Fundamentación de la didáctica*, México, Gernika, Tomo I, 6ª edición, pp. 180-196.

PLANIED: *Plan Integral de Educación Digital del Ministerio de Educación y Deportes de la Nación Argentina* (s.f). Recuperado de: https://bit.ly/2GQYi16. Último acceso: 19/02/2019.

Program.AR: programa de la Fundación Sadosky, Ministerio de Ciencia, Tecnología e Innovación Productiva de la Nación Argentina (s.f). Recuperado de: http://program.ar/. Último acceso: 14/04/2019.

Queiruga C., Banchoff Tzancoff C., Venosa P, Gómez S., Morandi G. (2019). Computer Science and Schools: a Specific Didactics?. In: Pesado P., Aciti C. (eds) Computer Science – CACIC 2018. CACIC 2018. Communications in Computer and Information Science, vol 995. Springer, Cham.

Secundaria 2030: *Transformar la Secundaria para transformar vidas* (s. f). Recuperado de: https://bit.ly/2XcmXmt. Último acceso: 19/02/2019

Zambrano Leal, A. (2016) “Pedagogía y didáctica: esbozo de las diferencias, tensiones y relaciones de dos campos”. En *Praxis & Saber, vol. 7, núm. 13,* Universidad Pedagógica y Tecnológica de Colombia. Disponible en: https://bit.ly/2Uhgh3F. Último acceso: 14/04/2019.

Análisis de producciones de docentes de Educación primaria en formación sobre Didáctica de las Ciencias de la Computación

Marcela Daniele[1], Teresa Quintero[2], Cecilia De Dominici[3],
Flavia Buffarini[4], Francisco Bavera[5]

Resumen: Como actividad de cierre e integración del primer año de dictado de la Especialización Docente de Nivel Superior en Didáctica de las Ciencias de la Computación, se llevaron a cabo las *I Jornadas de Experiencias Docentes en Didáctica de las Ciencias de la Computación*. En dichas jornadas, los cursantes docentes de nivel primario, presentaron y difundieron sus producciones y actividades realizadas, con el objetivo de integrar los conocimientos adquiridos durante el cursado de la Especialización en el año 2018. Este trabajo presenta resultados parciales a partir del análisis de dichos procesos de integración, presentados en formato póster por los estudiantes. El equipo de investigación analizó las presentaciones a través de una metodología cualitativa basada en la teoría fundamentada, y se obtuvieron datos que representan resultados preliminares que aportan al desarrollo de conocimientos sobre la formación continua de docentes de Educación Primaria en Didáctica de las Ciencias de la Computación y el impacto en sus aulas de la escuela primaria.

Palabras claves: Pensamiento Computacional, Habilidades Cognitivas, Integración, Formación docente continua.

1 Universidad Nacional de Río Cuarto, Facultad de Ciencias Exactas, Físico-Químicas y Naturales marcela@dc.exa.unrc.edu.ar
2 UNRC - Facultad de Ciencias Exactas, Físico-Químicas y Naturales, Instituto Superior Ramón Menéndez Pidal (Río Cuarto) tquintero@exa.unrc.edu.ar
3 Instituto Superior Ramón Menéndez Pidal (Río Cuarto) cdominici2565@gmail.com
4 UNRC - Facultad de Ciencias Exactas, Físico-Químicas y Naturales, Instituto Superior Ramón Menéndez Pidal (Río Cuarto) fbuffarini@exa.unrc.edu.ar
5 UNRC - Facultad de Ciencias Exactas, Físico-Químicas y Naturales pancho@dc.exa.unrc.edu.ar

1. Introducción (problema y objetivos)

La enseñanza de las Ciencias de la Computación permite desarrollar estrategias para la resolución de problemas, el razonamiento lógico, y el pensamiento computacional, habilidades de pensamiento de orden superior, planteado en la taxonomía revisada de Bloom para la era digital por Andrew Churches [1], quien adecua la taxonomía, complementando cada categoría con verbos y herramientas del mundo digital desarrollando habilidades para *recordar*, *comprender*, *aplicar*, *analizar*, *evaluar* y *crear*. El pensamiento computacional consiste en la resolución de problemas, el diseño de sistemas, la comprensión de la conducta y de las actividades humanas, haciendo uso de procedimientos básicos para la elaboración de programas y algoritmos informáticos, valiéndose para ello de habilidades específicas y técnicas necesarias, constituyendo la base de la cultura digital [2,9,10,11].
Es necesario abordar la formación de ciudadanos que no solo consuman tecnologías, sino que sean capaces de crearlas, dando sentido al lema aprender para programar, programar para aprender. Construir el pensamiento computacional en edades tempranas obedece a una necesidad social de este siglo, de resolver problemas utilizando la tecnología, pero también desarrollar la capacidad de diseñar y crear soluciones a problemas de diferentes tipos, aprovechando los beneficios que brinda la informática y la tecnología. Construir el pensamiento computacional y aprender la programación, se perfila como el camino para el desarrollo de estas habilidades en la primera infancia, tanto habilidades cognitivas como actitudinales. Usar la tecnología para aprender, y no solo ser usuarios, es el principal desafío para este siglo.

Sin dudas, el desarrollo del pensamiento computacional (PC) en la educación integral de ciudadanos críticos, presenta un desafío de relevancia para el sistema educativo. Estudios y tendencias a nivel mundial dan cuenta de la importancia de introducir el PC en el sistema educativo obligatorio. Estonia, Reino Unido, Finlandia, Francia y Australia han sido pioneros en esta decisión. En países de América Latina, como Costa Rica, Perú, Colombia y República Dominicana, han generado diversas propuestas e incluido las ciencias de la computación en sus sistemas educativos. En la Argentina, en el año 2015, el Consejo Federal de Educación, destacó la relevancia que reviste en la actualidad la enseñanza y el aprendizaje significativo de la programación "la programación es de importancia estratégica en el Sistema Educativo Nacional durante la escolaridad obligatoria, para fortalecer el desarrollo económico-social de la Nación, conforme lo establecido por el artículo 3º de la Ley de Educación Nacional" (Resolución N° 263/15). Y en 2018, este consejo aprobó los núcleos de aprendizaje prioritarios para la educación digital, programación y robótica (NAP, Res. CFE Nº 343/18), imponiendo a las jurisdicciones analizar y llevar adelante su implementación e inclusión en sus documentos curriculares.

La construcción del pensamiento computacional requiere mayor análisis e investigación, explorando sus dimensiones, importancia y beneficios [8]. En la actualidad, es escaso el número de tesis o artículos científicos acerca del tema. En nuestro país, y en particular en nuestra provincia, se hace necesario profundizar en trabajos de investigación que reflejen

los resultados obtenidos con educadores, como así también, con alumnos de la escuela primaria, que hayan recibido formación apuntada a la construcción del pensamiento computacional. En la provincia de Córdoba, existen algunas escuelas que tienen una orientación hacia la informática, algunas en TICs, como así también otras en programación. Además, en el año 2014, comenzaron a funcionar las escuelas experimentales PROA (Programa Avanzado en Educación) haciendo foco en las TICs y en el desarrollo de software. Se trata de un proceso innovador para escuelas secundarias, donde la provincia es la primera en Argentina en implementar este tipo de formato educativo que tiene como objetivo la formación en tecnologías de la información y comunicación, otorgando el título de bachiller en desarrollo de software. Por otra parte, la Fundación Dr. Manuel Sadosky, dependiente de la Presidencia de la Nación Argentina, comenzó a desarrollar el programa Program.AR [3] en el año 2013, introduciendo en algunas comunidades educativas de las escuelas de todo el país, nociones de programación y robótica, así como la didáctica de las ciencias de la computación.

En la actualidad el desafío es evaluar en profundidad el impacto de estas iniciativas en la escuela y en la formación de los docentes [8], en particular, estudiar si a partir de estas experiencias los docentes construyen dialécticamente su propio pensamiento computacional y se apropian de *maneras/modos* de construcción que se constituyen en nuevas prácticas educativas que determinan la posibilidad del desarrollo de dicho pensamiento en sus estudiantes de la escuela. Este equipo de investigación, ha comenzado un proceso de estudio intentando atrapar parte de la complejidad inherente al proceso de construcción del pensamiento computacional en la escuela, que se identifica a partir de los ejemplos particulares que representa cada grupo -docentes y sus estudiantes de la escuela- en los que se focaliza la investigación. La producción de conocimiento de esta investigación será uno de los insumos para el desarrollo de propuestas de enseñanza situadas para la formación y el desarrollo del pensamiento computacional en docentes y estudiantes.

Como actividad de cierre e integración del primer año de dictado de la Especialización Docente de Nivel Superior en Didáctica de las Ciencias de la Computación, iniciativa conjunta de la Fundación Sadosky, el Instituto Superior de Formación Docente Ramón Menéndez Pidal, y la Facultad de Ciencias Exactas Físico-Químicas y Naturales, UNRC, se llevaron a cabo las *I Jornadas de Experiencias Docentes en Didáctica de las Ciencias de la Computación*. En dichas jornadas, los cursantes docentes de Nivel Primario, presentaron y difundieron sus producciones y actividades realizadas, reflejando la integración de los conocimientos adquiridos durante el cursado de la Especialización, en el año 2018.

Este trabajo, refleja la investigación realizada a partir del análisis de los productos de integración, presentados en formato póster por los estudiantes. El equipo de investigación, realizó un análisis de cada presentación usando una metodología cualitativa, basada en la Teoría Fundamentada [6,7], y se obtuvieron datos que representan resultados preliminares que aportan al desarrollo de conocimientos sobre la formación continua de docentes de Educación Primaria en Didáctica de las Ciencias de la Computación y el impacto en sus aulas de la escuela primaria.

2. Descripción de la Propuesta

La *I Jornada de Experiencias Docentes en Didáctica de las Ciencias de la Computación*, se llevó a cabo el día 24 de noviembre de 2018 en el Instituto Superior de Formación Docente Ramón Menéndez Pidal de la ciudad de Río Cuarto, constituyendo un espacio de integración de conocimientos y producciones realizadas por los profesores de la escuela primaria durante el primer año de cursado de la Especialización Docente de Nivel Superior en Didáctica de las Ciencias de la Computación. La muestra de producciones, realizadas en el marco de esta primera jornada de integración y sociabilización de experiencias, se realizó en modalidad de póster.

2.1. Los docentes participantes

Participaron como autores, cuarenta y tres (43) Profesores de Educación Primaria (40 maestras y 3 maestros), estudiantes de la especialidad, que forman parte de veintiocho (28) escuelas primarias de la ciudad y la región. El 98% de estos docentes de nivel primario son maestras/os de grado y sólo una maestra del área Informática. En el gráfico Nº 1 se presenta más detalles sobre las especialidades de la formación de estos docentes. Y en el gráfico Nº 2 se puede ver las asignaturas que están a su cargo.

La *media* de edad es 42 años, en un rango que va desde los 23 a los 54 años. Con una *mediana* y una *moda* de 43 y 44 años, respectivamente. La *desviación media* es de 4,5. El 50% son docentes titulares y los restantes son interinos y suplentes, que se desempeñan en los diferentes grados de la escuela primaria. Además, dos docentes no están relacionados con institución alguna y un docente desempeña actividades en forma ad-honorem. Como se mencionó previamente, los docentes pertenecen a 28 escuelas, donde el 80% son instituciones públicas de gestión estatal y el restante 20% instituciones públicas de gestión privada. Un 75% de estas escuelas están localizadas en la ciudad de Río Cuarto, y el restante 25% en localidades de su región de influencia (en un radio de 30 a 150 km de distancia de Río Cuarto).

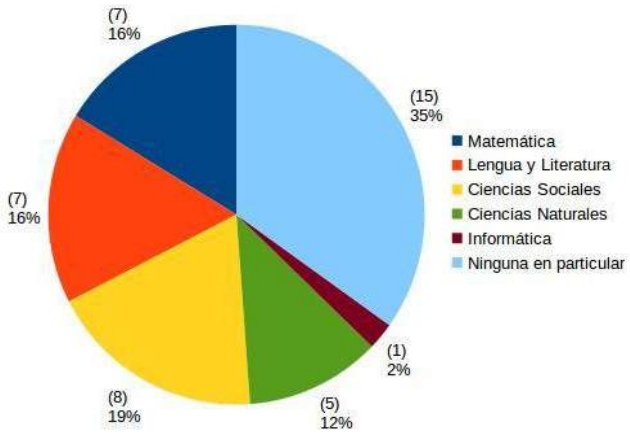


Gráfico N°1. Especialidad en la Formación de los docentes cursantes.

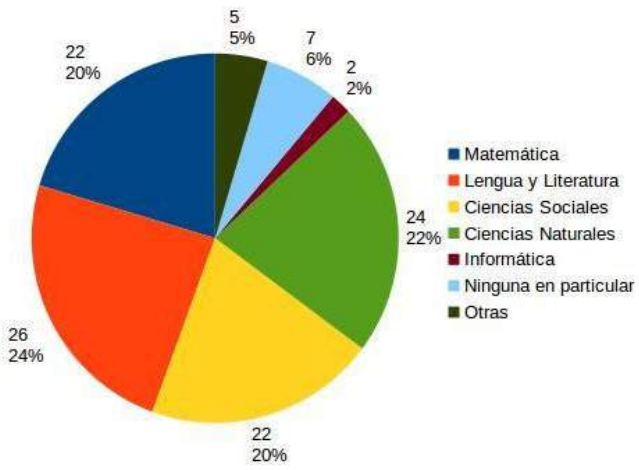


Gráfico N°2. Materias dictadas por los docentes cursantes.

El 98% de los docentes informaron, al comienzo de la especialidad, que tenían conocimientos de informática. De los cuales, el 30% proviene de capacitaciones formales, otro 30% recibió capacitaciones informales y el restante 38% se considera autodidacta. Ninguno participó previamente en capacitaciones relacionadas con las Ciencias de la Computación, la programación o la robótica. El 2% restante no tiene conocimiento alguno de informática.

Al indagar sobre las experiencias previas que tuvieron con la informática en el ejercicio docente se pudo recabar que 21 docentes emplearon utilitarios de oficina, 7 aplicaciones de celulares y 9 docentes otras aplicaciones. Mientras que 12 maestras nunca realizaron experiencias relacionadas con la informática en su aula. Sólo 5 de los docentes tuvieron alguna experiencia con lenguajes de programación visuales y 2 con robótica educativa.

2.2. La Especialización

La Especialización tiene una duración de 400 horas, divididas en 8 módulos, a lo largo de dos años. Los distintos módulos introducen a los docentes en conceptos de ciencias de la computación, ciudadanía digital, pensamiento computacional, programación y robótica. Durante el primer año se dictaron 4 módulos: (1) Herramientas de comunicación y colaboración, (2) Introducción a la Resolución de problemas, (3) Introducción a los Lenguajes de Programación y (4) Administración y Configuración de Software y Hardware, y en forma transversal se acreditaron horas en el módulo Práctica Docente. Culminando, este primer año, con la *I Jornada de Experiencias Docentes en Didáctica de las Ciencias de la Computación*. La evaluación de cada módulo estuvo centrada en actividades y experiencias áulicas, generadas e implementadas en forma grupal por los docentes, a modo de actividades integradoras en cada uno de ellos.

Los temas vistos en estos módulos incluyen: huella digital, ciudadanía digital responsable, derechos de autor y propiedad intelectual. Estrategias para la resolución de problemas, abstracción, reconocimiento de patrones y generalización. Instrucciones, algoritmos, lenguajes de programación. Prueba y depuración de algoritmos. Verificación y validación de algoritmos. Relación entre software y hardware. Conceptos básicos de sistemas operativos y redes.

A lo largo del dictado se utilizaron actividades enchufadas y desenchufadas. Entre las herramientas utilizadas en las actividades enchufadas se pueden mencionar: Lightbot, code.org, Pilas-Bloques, Tortugarte y m-bot.

3. Metodología y Análisis de Resultados

Se planteó una investigación exploratoria, descriptiva desde un enfoque cualitativo. Para el análisis de las producciones se siguió una metodología cualitativa, basada en la Teoría Fundamentada. Los datos obtenidos fueron triangulados entre sí, por al menos dos investigadores. Los objetos de estudio fueron los posters de integración presentados por los estudiantes de la especialidad, a saber, docentes de educación primaria en actividad. Se analizaron los 15 (quince) posters presentados y se consideraron las notas tomadas por los investigadores en las presentaciones de los mismos en la *I Jornada de Experiencias Docentes en Didáctica de las Ciencias de la Computación*. Los mismos se analizaron en función de las habilidades de pensamiento según la Taxonomía de Bloom para la era Digital [1]:

- Crear. Verbos asociados: diseñar, construir, planear, producir, idear, trazar, elaborar, programar, filmar, animar, participar en una wiki, publicar, dirigir, transmitir, transferir.
- Evaluar. Verbos asociados: revisar, formular hipótesis, criticar, moderar, colaborar, experimentar, juzgar, probar, detectar, monitorear, comentar en un blog, participar en redes, reelaborar.
- Analizar. Verbos asociados: comparar, organizar, deconstruir, atribuir, delinear, encontrar, estructurar, integrar, recombinar, enlazar, validar, recopilar información.
- Aplicar. Verbos asociados: implementar, desempeñar, usar, ejecutar, correr, cargar, jugar, operar, subir archivos a un servidor, compartir, editar.
- Comprender. Verbos asociados: interpretar, resumir, inferir, parafrasear, clasificar, comparar, explicar, ejemplificar, hacer búsquedas avanzadas, categorizar, etiquetar, comentar, anotar.
- Recordar. Reconocer, listar, describir, identificar, recuperar, denominar, localizar, encontrar, utilizar viñetas, resaltar, marcar, participar en la red social, buscar en google.

3.1. Las producciones integradoras

Se desarrollaron un total de quince (15) producciones (Tabla N° 1), donde las y los formadores de distintas escuelas se agruparon y elaboraron las diferentes propuestas.

Número del trabajo	Título del trabajo
1	Nuevos lenguajes
2	Aprender conectados
3	Genios de la Programación
4	Nuevos aprendizajes
5	El pensamiento computacional: un desafío para la escuela actual
6	Posibilidades y desafíos hacia nuevos horizontes
7	¿Razonamos juntos?
8	Estrategias matemáticas para el universo tecnológico
9	Ciencias de la Computación y su didáctica
10	Nuestro recorrido
11	Entramados en las Ciencias de la Computación
12	El pensamiento computacional a las aulas. Diseñando un futuro
13	Creamos, pensamos y comunicamos
14	Culturalmente robotizados
15	Desde la primera infancia: programación activa y no consumo pasivo

Tabla N°1. Títulos de las producciones presentadas en la I Jornada de Experiencias Docentes en Didáctica de las Ciencias de la Computación.

Las producciones representaron diferentes dimensiones de la formación docente: cómo habitaron la formación y capacitación sobre un contenido disciplinar, cómo construyeron conocimientos y cómo resolvieron la transposición didáctica en sus aulas de la escuela primaria.

Todos los equipos de trabajo, diseñaron estructuras gráficas para representar la forma en que integraron los conocimientos a la formación docente inicial y a sus prácticas áulicas. Uno de los equipos, describe el recorrido en *forma espiralada*, dando cuenta de una construcción dialéctica y reflexiva del conocimiento. "... representa el crecimiento en nuestros conocimientos, como así también, en las experiencias áulicas realizadas. Reflexionamos sobre cómo este recorrido modificó nuestras prácticas docentes y cómo resignificamos y adaptamos los contenidos y actividades áulicas a partir de nuestro recorrido

por el postítulo…" (Equipo 10: *Nuestro Recorrido*). Otro grupo (Equipo 7), lo representa con figuras geométricas, dando cuenta de una construcción integrada y progresiva de los conocimientos. "... la jerarquización del círculo central, se debe a que la enseñanza en los distintos espacios curriculares centrada en el desarrollo de las capacidades fundamentales prioritarias permite a todos los estudiantes participar y potenciar las mismas de acuerdo a sus posibilidades…" (Equipo 7: *¿Razonamos Juntos?*).

La mayoría de los trabajos, se centraron en la formación de Ciudadanos Críticos, "... En el óvalo central se ubicó el Ciudadano Digital, debido a que se pensó que el punto de partida es promover el desarrollo de ciudadanos críticos, responsables , libres e integrados, en el mundo real y digital…" (Equipo 14: *Culturalmente Robotizados*), en el sentido de reconocer la enseñanza de las Ciencias de la Computación en el Nivel Primario con el objetivo de "... formar ciudadanos con conocimientos suficientes sobre tecnología que les permitan participar en debates, con libertad de expresión en la red, capaces de comprender cómo funciona, capaces de intervenir y capaces de crear…" (Equipo 4: *Nuevos Aprendizajes*).

Otros equipos integraron los contenidos de los diferentes módulos de aprendizajes, a partir de resaltar la relevancia de los conceptos disciplinares y su relación con el desarrollo del pensamiento infantil. "... La enseñanza de la programación fomenta el pensamiento computacional. Este proceso de razonamiento estructura la mente y ordena las ideas, permitiendo dividir un problema grande en problemas más pequeños, hasta encontrar la solución..." (Equipo 7: *¿Razonamos Juntos?*).

La mayoría de los trabajos resaltan las relaciones que establecieron entre la formación sobre un nuevo contenido disciplinar y las formas de integrarlos al aula y a sus prácticas docentes. El Equipo 15, propone incorporar los conceptos de las Ciencias de la Computación de manera "transversal", para procurar "... un rol activo el del docente en la formación de los ciudadanos digitales desde la primera infancia…" (Equipo 15: *Desde la primera infancia: programación activa y no consumo pasivoteórico*). En el mismo sentido el Equipo 2, plantea un rol activo del docente en relación a la incorporación de los contenidos en el aula, anticipando la participación del mismo en la evaluación crítica de los recursos y entornos digitales para "... problematizar e interpretar situaciones reales, a la vez que desarrollar la creatividad y el trabajo en colaboración, como así también contribuye al acercamiento a la tecnología desde un posicionamiento más crítico…" (Equipo 2: *Aprender Conectados*). En el Póster *Genios de la Programación*, el Equipo 3, plantea la necesidad de realizar una "transposición didáctica" para el inicio de la construcción de conocimientos y como "... forma de ampliar el horizonte de nuestras prácticas y los aprendizajes mutuos en esta triada tan conocida de estudiante-conocimiento-docente…" (Equipo 3: *Genios de la Programación*).

Algunas habilidades que se encontraron en las producciones y que se identifican con las del pensamiento en la era digital, son: dar y seguir instrucciones, secuenciar, ordenar, probar, ejecutar, verificar, analizar, validar, codificar, validar distintas maneras de pensar, identificar situaciones, descomponer en subproblemas, construir algoritmos, reconocer patrones, usar

herramientas de colaboración y comunicación, aplicar estrategias, trabajar colaborativamente en documentos compartidos, wikis, redes sociales, buscar información en la red, abordaje y resolución de problemas, programación en acción, robótica, producciones creativas en documentos compartidos, fundamentando procedimientos. En la Tabla N° 2 se consignan las producciones en función de las habilidades de pensamiento detectadas en los mismos.

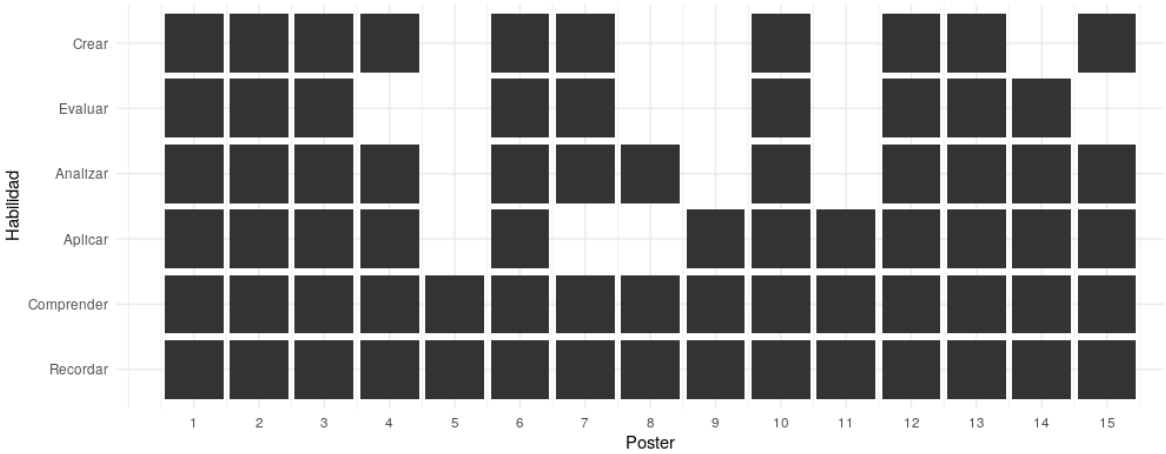


Tabla N°2. Habilidades de pensamiento detectadas en las producciones.

Del análisis de las producciones y de las presentaciones realizadas, podemos expresar que los docentes han logrado trabajar contenidos y habilidades abordados en los módulos de la Especialización en distintas áreas de la enseñanza primaria, aplicación en otras áreas escolares (ciencias naturales, ciencias sociales, matemática y lengua).

En la mayoría de los trabajos se observa, la transferencia al aula y el trabajo con sus estudiantes de algunas de las propuestas desarrolladas en la especialización. Como ejes trabajados podemos decir que las producciones, giraron principalmente sobre: situaciones problemáticas, desarrollo del pensamiento computacional, trabajo colaborativo y ciudadanía digital.

Los trabajos en los que se identificó la presencia de las seis habilidades tienen en común un fuerte trabajo en la búsqueda y explicitación de estrategias en la resolución de los problemas con posteriores debates y análisis entre sus alumnos sobre las distintas soluciones. Hicieron hincapié en la validación de las soluciones, la detección y solución de los errores. Trabajaron claramente sobre la necesidad de contar con instrucciones precisas y no ambiguas en la definición de secuencias de instrucciones (algoritmos) para expresar las soluciones a los problemas. Además, en todos estos trabajos se implementaron actividades enchufadas.

Uno de los grupos de docentes (Equipo 6), avanzó más allá y junto con sus estudiantes, participó de una actividad intercambiando experiencias de informática en la escuela, e*l*

Primer encuentro PracTIC.ar: Viralizando la educación digital. Además, los docentes del Equipo 10 presentaron su experiencia, en formato póster, en las II Jornadas Argentinas de Didáctica de la Programación (JADiPro 2019).

4. Conclusiones

En este primer análisis encontramos que todas las producciones y las propuestas para sus prácticas áulicas, reflejaron habilidades cognitivas relacionadas al mundo digital, mostrando diferentes niveles de desarrollo y profundidad. Desde este trabajo exploratorio e inicial, se intenta aportar al desarrollo de conocimientos sobre la formación continua de docentes de Educación Primaria en Didáctica de las Ciencias de la Computación y el impacto en sus aulas de la escuela primaria. Esta es una primera aproximación y se continuará trabajando en esta línea de análisis, que nos posibilitará generar conocimientos, para ir adecuando e integrando las propuestas desarrolladas a nuevas versiones o réplicas de cursos de Formación Docente.

Se pretende continuar valorando las habilidades de Pensamiento Computacional que poseen los docentes participantes de la Especialización, con otras metodologías. También se está trabajando en la evaluación de la inclusión del PC en la Formación Docente Inicial y en la colaboración con los docentes que cursaron la Especialización en el diseño e implementación de propuestas didácticas en aulas de las Escuelas Primarias.

Por otra parte, se planifica incursionar en el análisis de las percepciones de los docentes de Educación Primaria con respecto a las Ciencias de la Computación, la programación y la robótica. Este análisis también pretende vislumbrar el aporte, en cuanto al cambio de las percepciones de los docentes, de distintas formaciones en Ciencias de la Computación, Pensamiento Computacional, Programación y Robótica que se están llevando a cabo.

5. Agradecimientos

Este trabajo es apoyado en parte por el Ministerio de Ciencia y Tecnología de la Provincia de Córdoba (Argentina) a través del Proyecto de Investigación y Desarrollo (PID 2019-2020) *La construcción del pensamiento computacional: estudio del impacto de la formación docente*. Se agradece la financiación del UNRC para el proyecto (PPI 2020-2022) *Pensamiento computacional y prácticas de enseñanza en ciencias*. Queremos agradecer a las y los docentes de Nivel Primario que participaron de la Especialización de Enseñanza Superior en Didáctica de las Ciencias de la Computación, a la Fundación Sadosky y al Instituto Ramón Menéndez Pidal. También un especial agradecimiento a nuestros colegas que participaron del dictado de la Especialización que proporcionaron información y experiencias que ayudaron en gran medida a la investigación. Finalmente, agradecemos a los revisores de las JADiPro 2019 por sus sugerencias y aportes a este trabajo.

Bibliografía

Churches, A. (2009). *Taxonomía de Bloom para la era digital*. Eduteka. Consultado 10/01/19. Disponible en: http://eduteka.icesi.edu.co/articulos/TaxonomiaBloomDigital

Zapata-Ros, M. (2015). "Pensamiento computacional: Una nueva alfabetización digital". En *RED-Revista de Educación a Distancia*. Número 46. Consultado el 21/11/18. Disponible en http://www.um.es/ead/red/46

Program.AR, Disponible en: http://program.ar/

Anderson, L.W., y Krathwohl, D. (2001). *A Taxonomy for Learning, Teaching and Assessing: a Revision of Bloom's Taxonomy of Educational Objectives*. Longman, New York.

Churches, A. (2007). Educational Origami, Bloom's and ICT Tools http://edorigami.wikispaces.com/Bloom's+and+ICT+tools.

Strauss, A., Corbin, J. (2002). *Bases de la Investigación Cualitativa. Técnicas y procedimientos para desarrollar la Teoría Fundamentada*. Medellín (Colombia): Editorial Universidad de Antioquia.

Alarcon, A., Múnera Cavadías, L., Montes Miranda, A. (2016). "La teoría fundamentada en el marco de la investigación educativa". En *Revista Saber ciencia y libertad.* 12(1):236-245. DOI: 10.18041/2382-3240/saber.2017v12n1.1475

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K. (2016). *Developing computational thinking in compulsory education – Implications for policy and practice*; EUR 28295 EN; doi: 10.2791/792158

Aho, A. V. (2012), *Computation and Computational Thinking. The Computer Journal*, 55(7), 832-835. doi: https://doi.org/10.1093/comjnl/bxs074.

Denning, P. J. (2017), Remaining trouble spots with computational thinking. Communications of the ACM, 60(6), 33-39. doi: https://doi.org/10.1145/2998438.

Wing, J. M. (2006), Computational thinking. Communications of the ACM, 49(3), 33-35. doi: https://doi.org/10.1145/1118178.1118215.

La enseñanza de la programación a través del ABP. Una experiencia de formación en el Profesorado de Informática

Silvia Pilar Rodríguez[1]

Resumen: Formar un docente de programación implica, en primer lugar, establecer estrategias que aseguren la correcta comprensión de la resolución de algoritmos para su posterior trasposición didáctica. En este sentido, no se persiguen los mismos objetivos que al formar un técnico en programación cuya principal tarea será resolver problemas. En el caso de los futuros docentes, es indispensable poner el acento en el modo en que se llega a esa solución. Sin embargo, en ocasiones algunos profesorados pueden olvidar este costado de la enseñanza. El Aprendizaje Basado en Proyectos es un aliado privilegiado y puede ser utilizado a la hora de formar al docente, no sólo explicando en qué consiste la metodología sino fundamentalmente utilizándola en la enseñanza disciplinar. Se trata de una "metaenseñanza" de la programación y su didáctica a través del ABP. El objetivo de este espacio de trabajo dentro de las jornadas es reflexionar sobre la cuestión didáctica en la formación de formadores, aspecto en general bastante desatendido. A partir de una metodología de trabajo concreta y específica, se buscará mostrar la importancia vital que tiene no descuidar este aspecto en el nivel superior.

Palabras clave: programación, didáctica, docentes, profesorado, ABP.

1 I.S.P. "Dr. Joaquin V. González (CABA).

Introducción

La programación como asignatura escolar es relativamente nueva, y por eso su didáctica no está suficientemente desarrollada aún. Son vitales, por tanto, estos espacios de reflexión en los que podamos intercambiar vivencias y enriquecernos. La que se relata a continuación es una experiencia desarrollada en el Profesorado de Informática del Instituto Superior del Profesorado "Joaquín V. González" de la Ciudad Autónoma de Buenos Aires.

En el mencionado instituto se les enseña a los futuros docentes varias técnicas de programación y se codifica en una buena cantidad de lenguajes. También existe una batería de espacios curriculares que dan cuenta de la formación general del profesorado basada en la psicología, la pedagogía y la didáctica. Si bien existe una relación cordial y profesional entre los docentes de los distintos espacios tanto generales como específicos, en varias oportunidades se ha escuchado a los alumnos manifestar su preocupación al advertir ciertas inconsistencias entre ambos ejes. La formación disciplinar no parece contemplar los aspectos pedagógicos y didácticos con los que los estudiantes se encuentran en el eje de la formación docente. Esta situación provoca que luego los futuros profesores tiendan a repetir los modelos disciplinares de enseñanza, que en muchos casos se orientan más a una formación de tipo técnico de resolución de problemas, que a poner en un lugar central la idea de la transmisión futura que deberá hacerse de ese conocimiento. Después de todo y como señala Andrea Alliaud, un docente que se enfrenta ante una clase por primera vez, no lo hace como inexperto; al contrario, cuenta con varios años de experiencia de alumno que suele tener un peso difícil de contrarrestar con la sola transmisión de saberes acerca de modelos didácticos [CITATION All04 \t \l 11274], mucho más si a eso le agregamos que la propia formación del profesorado la transitan con estos modelos.

Tal vez en este punto sea dable destacar que no se trata de una falencia sólo de nuestra disciplina. Por el contrario, son varias las que evidencian este mismo divorcio formación general – formación específica. Los dichos de los alumnos pueden corroborarse en los perfiles de las personas que trabajan en los institutos de formación docente: muchos de los profesores que dictan materias disciplinares tienen títulos profesionales y no docentes. Así encontramos arquitectos a cargo de la materia Diseño Gráfico, ingenieros agrónomos que dictan clases de Utilitarios, o ingenieros en sistemas que se ocupan de las cátedras de Programación. Todos estos profesionales, en su formación disciplinar en licenciaturas, ingenierías o tecnicaturas, transitaron un aprendizaje cuyo objetivo era la ejecución de un sistema en sí mismo, y no la manera en que este aprendizaje debía transformarse en un saber que pudiera a su vez ser transmitido. ¿Puede un profesional de la informática, formado en esta lógica utilitarista, enseñar programación con una mirada didáctica que apunte a las formas de transmisión del saber más que al contenido? Claramente sí, no tenemos por qué pensar que hay algún impedimento per se. No obstante, tampoco podemos esperar que lo hagan espontáneamente si no hay alguna capacitación o al menos, alguna orientación al respecto.

En atención a esta circunstancia que nos preocupa y que estamos observando en nuestro profesorado, sumado a los dichos de los alumnos que corroboran la situación, se llevó a cabo una experiencia de formación disciplinar en la que se puso el énfasis más en la didáctica de la programación que en la técnica. Es decir, no sólo se les enseñaron las estrategias de resolución de algoritmos y codificación, sino que se hizo hincapié en la forma de construir ese conocimiento, se los hizo conscientes del modo en que ellos estaban aprendiendo. De esta manera, los estudiantes vivenciaron una metodología de trabajo a través de la cual aprendieron una técnica de programación dentro de parámetros didácticos aplicables a alumnos de nivel medio, los futuros destinatarios de su actividad.

Por otra parte, un escenario también bastante frecuente en la formación superior, es la subestimación de la importancia de la didáctica para el éxito de las trayectorias académicas. Se presume que en estos niveles de formación no es necesario atender este aspecto: los estudiantes son adultos y por lo tanto autónomos, eso nos lleva a la conclusión de que el profesor de nivel superior no necesita elaborar distintas estrategias de enseñanza para que sus alumnos aprendan efectivamente; es suficiente que haya un docente que transmita para que los alumnos construyan conocimiento [CITATION Sta12 \l 11274]. Además de que esta premisa no se sostiene en la práctica (basta advertir el desgranamiento que se da en los niveles superiores, lo que da en cuenta en parte de una imposibilidad de seguir los ritmos institucionales impuestos), autores como Mariana Maggio ponen en evidencia la obvia refutación de esta hipótesis [CITATION Mag18 \l 11274]. La autora del libro “Reinventar la clase en la universidad” dedica buena parte de su investigación a puntualizar la urgente necesidad de repensar las prácticas docentes en el nivel superior, a fin de evitar fracasos y deserciones que imposibiliten aspirar a un escenario de igualdad de oportunidades.

Pero este no es el único motivo por el cual es indispensable trabajar en una lógica que haga hincapié en la metodología didáctica. Es necesario destacar, como ya hemos dicho anteriormente, que no puede compararse la formación de un técnico o de un profesional con la formación de un docente. Es decir, el objetivo último del aprendizaje no es el contenido en sí mismo sino fundamentalmente su forma de transmisión. Por eso es forzoso que el alumno -futuro docente- sea capaz de hacer conscientes las estrategias empleadas en la construcción del conocimiento, evidenciando cada paso de crecimiento en el logro de los objetivos. De lo contrario, no podrá hacer la transposición didáctica requerida a la hora de cumplir con su tarea docente y no hará más que repetir modelos tradicionales de enseñanza largamente fracasados. En este sentido, la presente experiencia apostó fuertemente al metaaprendizaje, reflexionando sobre cada uno de los avances.

Descripción de la experiencia

La asignatura donde se llevó a cabo esta experiencia se denomina “Proyectos Educativos”, perteneciente al 4to. año de estudios, y tiene como objetivo, entre otros, que los futuros docentes diseñen e implementen proyectos informáticos orientados a lo educativo, aplicando las técnicas de programación que se estudiaron en materias anteriores (Programación I, II y

III). Es decir, deben diseñar un sistema informático que involucre una serie de conocimientos determinados, y cuyo objetivo sea la utilización concreta en las instituciones educativas para la enseñanza de alguna disciplina, con las formalidades que ello implica: fundamentación, objetivos, tiempos, solicitud de recursos, etc. Se trata de una asignatura anual que forma parte del nuevo plan de estudios, aprobado en el año 2015, por lo que la primera vez que se dictó fue en 2018 con seis alumnos. La experiencia se repitió hasta el presente en los siguientes cursos en los que se sigue adoptando la misma metodología.

Una de las maneras de llevar adelante la tarea que implica este espacio curricular, es indicarles a los estudiantes cómo se diseña un proyecto didáctico, cuáles son las distintas etapas del mismo, cómo se estructuran los objetivos, de qué modo se incluyen los contenidos, en fin, todos los aspectos que hacen a la planificación de proyectos. Para que lleguen a elaborar ese proyecto podríamos introducirlos en algunos aspectos teóricos básicos que deben regir ese diseño.

Paralelamente, y porque así está establecido en el diseño curricular, deberemos perseguir como objetivo que aprendan a diseñar un sistema informático con todos sus ciclos de vida y puedan elaborar la correspondiente documentación respaldatoria de todo lo actuado, en un lenguaje normado y estandarizado. En este caso, deberíamos echar mano de variada bibliografía que da marco a esta actividad, y trabajarla en las clases para que luego puedan llevar a cabo la tarea.

Lo descripto en los párrafos anteriores es lo que llamaríamos “enseñanza tradicional” basada en tres etapas fundamentales: teoría, práctica y evaluación [CITATION Fel10 \l 11274]. Una serie de conceptos teóricos que se les muestra para que vean “cómo” se hace y “por qué”, una serie de ejercicios en los que puedan “aplicar” esos conocimientos, y finalmente, cuando el mecanismo está debidamente aceitado, una evaluación que en este caso podría ser la elaboración del proyecto educativo y el diseño del sistema informático.

A la hora de planificar el recorrido del espacio curricular de nivel superior, se detectó la necesidad de generar alguna inquietud en los futuros docentes, quienes por lo anteriormente expuesto, tienden a concebir una única manera de “dar clase” [CITATION Ede11 \l 11274]. Para ello, se dedicó el primer encuentro a conversar acerca de visiones de enseñanza y de aprendizaje, representaciones sociales de lo que significa la programación de sistemas, concepciones del lugar que ocupa la informática en la escuela, etc. Se realizó un punteo con los dichos que ellos vertían y al encuentro siguiente se trabajó cada uno de ellos, evidenciando las distintas concepciones pedagógicas y didácticas que subyacen a cada afirmación surgida. La intención de esta actividad fue hacer conscientes las representaciones e imaginarios que se tienen acerca de lo que es “enseñar” y “aprender”, en todo el amplio espectro de significados de ambos procesos. Para ello se utilizó la estrategia didáctica del cambio conceptual [CITATION Dav11 \l 11274], enfrentando a los estudiantes a sus propias manifestaciones, poniendo en conflicto las mismas con las teorías de enseñanza que estudian en las materias que componen el eje de formación docente. La respuesta no se hizo esperar: ante una de las frases que se explicitaron uno de ellos

exclamó: “¿¡Nosotros dijimos eso!?”. Claramente se habían hecho conscientes las representaciones acerca de los procesos de enseñanza y aprendizaje que habían ido formando a lo largo de toda su biografía escolar, y que distaban mucho de las teorías que se estudian en el profesorado.

Varias fueron las clases que se dedicaron a reflexionar sobre estos temas: era fundamental que los alumnos generaran en ellos mismos, la necesidad de ejercer una práctica reflexiva [CITATION Bru00 \l 11274] que los llevara a cuestionarse sus propias concepciones de enseñanza y aprendizaje. Se les propuso entonces trabajar con una metodología didáctica que era la misma que ellos debían concebir para su propuesta: el Aprendizaje Basado en Proyectos (ABP). Primeramente se los invitó a diseñar un proyecto, a partir del cual, mediante la metodología de diálogo reflexivo [CITATION Dav11 \l 11274], pudieron inducir la diferencia entre proponer un proyecto a los alumnos y enseñar un tema mediante ABP [CITATION del18 \l 11274]. De este modo, nos proponíamos quebrar esa lógica que ellos critican y que tiene que ver con el divorcio del que hablábamos entre formación docente y formación disciplinar.

Desde este punto de partida se les propuso entonces una nueva modalidad de trabajo, más allá de lo que específicamente indica el diseño curricular: ellos comenzarían a diseñar un proyecto áulico de programación en grupos de dos o tres integrantes, poniéndose en el lugar de profesores de distintos talleres: diseño, programación, manejo de aplicaciones, lenguajes de representación, etc. Cada uno asumiría un rol dentro de ese cuerpo docente imaginario (profesor de diseño, o de programación, etc.) y participaría de un proyecto en el que sus alumnos (compartidos con los demás profesores) colaborarían en la producción de un sistema informático. De este modo, un solo proyecto podía ser trabajado por los alumnos desde cada perspectiva con el profesor correspondiente. Desde la cátedra, se ocuparía el rol de director de la escuela o coordinador del área, que evaluaría la pertinencia o no del proyecto, solicitando los ajustes correspondientes. Rápidamente se organizaron y comenzaron a trabajar, pensando en contenidos, tiempos, recursos y demás elementos que ellos asumían que debían incluir.

La primera etapa del año se focalizó en el diseño del proyecto didáctico. En este caso, realizaron una propuesta para la cual los alumnos debían diseñar y programar un sitio web destinado a alertar sobre distintos casos de delitos en las redes sociales (acoso, distribución de datos no autorizados, suplantación de identidad, etc.). Trabajaron entonces sobre las distintas etapas que el proyecto iba a incluir, investigaron sobre la fundamentación de esta metodología de enseñanza, indagaron en las posibilidades de llevar adelante una tarea de este estilo, analizaron las alternativas de solución que aportaba cada lenguaje de programación, y finalmente redactaron el proyecto con pautas precisas de escritura y formato. Con el transcurrir del año, se les fueron haciendo observaciones, planteos, sugerencias, en orden a que afinaran su proyecto didáctico. De este modo, en el hacer mismo, aprendieron cómo se redacta y se presenta un proyecto.

Una vez que estuvo aprobado, el grupo comenzó a trabajar efectivamente en la elaboración del sistema, instancia durante la cual se los fue asesorando en cada una de las etapas, al punto tal que en algunos casos fue necesario volver sobre el proyecto original para modificarlo, dado que ellos mismos se encontraban con dificultades que permitían anticipar que no sería un proyecto viable. Tengamos en cuenta, en este punto, que la ejecución se hacía sobre las pautas que ellos mismos habían estado diseñando.

Este fue el momento del aprendizaje disciplinar de la programación. Si bien los estudiantes a esta altura de la carrera tienen un bagaje importante en torno al tema, el hecho de encontrarse en la situación de tener que resolver problemas concretos de un sistema, generó nuevos aprendizajes que llevaron a la articulación y a la integración de saberes previamente adquiridos. Así, por ejemplo, construyeron criterios de búsqueda de soluciones que les permitieron escalar en autonomía, seleccionaron el lenguaje de alto nivel más adecuado para el escenario en juego, elaboraron pautas de diseño de bases de datos para los propósitos iniciales, coordinaron estrategias de selección de alternativas de soluciones informáticas según el problema planteado… en suma, los aprendizajes en el área disciplinar adquiridos en otros momentos académicos por parte de los estudiantes, fueron puestos en juego en un trabajo colaborativo y de resolución de problemas concretos, lo cual constituyó en sí mismo un nuevo aprendizaje, además de una revitalización de los conceptos de programación ya estudiados.

Casi agotado el ciclo lectivo, dedicamos dos clases a generar la teoría que entre todos habían estado creando a partir de su experiencia práctica. En este punto resulta interesante hacer una observación particular.

Una persona dedica cuatro o cinco años de su vida a su formación docente inicial. Esto no significa que dicha formación se agotó en este tiempo. Muy por el contrario, en el ejercicio de la docencia el profesional deberá seguir capacitándose en las diversas ofertas académicas que se dan en las diferentes jurisdicciones. No obstante, esta heteroformación no es suficiente: el profesional reflexivo no sólo se nutre de teorías comprobadas y reflexiones de otros autores, sino que es autor de sus propias reflexiones que nacen de la práctica [CITATION Zei93 \l 11274]. Es por esto que se los invitó, durante todo el proceso, a elaborar una bitácora de recorrido en el que pudieron verter sus intercambios, sus idas y venidas, sus reflexiones. De este modo, ellos mismos llegaron a la conclusión de cuáles eran las pautas que se debían cumplir, cómo debían dirigirse a las autoridades escolares, qué debían tener en cuenta para decidir la viabilidad o no del proyecto, y demás cuestiones que no fue necesario “estudiarlas” sino que fueron internalizadas en la práctica y desde la práctica. [CITATION Aie11 \l 11274]

Nos parece vital formar docentes que no sean meros ejecutores de planes previamente diseñados y teorías elaboradas por otros, sino que por el contrario, sean profesionales críticos y productores de nuevas teorías a partir de estas reflexiones. Las aulas y las instituciones escolares están llenas de experiencias que merecen ser compartidas y analizadas críticamente para poner en juego nuevas estrategias de enseñanza surgidas

desde la realidad cotidiana de las escuelas. Pero está claro que esto es un ejercicio que no necesariamente se da de manera espontánea, sino que requiere de un entrenamiento, y sobre todo de una concientización sobre la importancia de encarar la tarea docente desde esta perspectiva. Es por eso que en este espacio curricular, este eje atravesó toda la experiencia.

Finalmente, observaron que la metodología con la que habían trabajado no era la tradicional sino que se enmarcaba en el Aprendizaje Basado en Proyectos, y que esta forma de enseñanza no es lo mismo que elaborar un proyecto para los alumnos. Así pudieron establecer la consigna básica del ABP que consiste en respetar los intereses de los alumnos, sus tiempos e incluso estar dispuestos, como docentes, a encontrarse con disparidad de contenidos y de niveles a los que los alumnos llegan. En resumen, estar dispuestos a gestionar un aula heterogénea. Pudieron comprobar de este modo, y desde la propia experiencia, que no sólo es posible llevar adelante esta metodología, sino que es más efectiva y más placentera.

Conclusiones

Los estudiantes manifestaron al final del curso que habían disfrutado su proceso de aprendizaje ("*lo pasamos re bien profe*"), valoraron la experiencia de una nueva forma de acercarse al conocimiento (*"nunca habíamos aprendido de esta manera"*), fueron capaces de reflexionar sobre la práctica y sintetizar teoría ("*me di cuenta de la diferencia, claro, es un nuevo método!*"), generaron un espacio de instalación del cuestionamiento permanente, revisitaron su biografía escolar, y lo que es más interesante, manifestaron su ansiedad por poner en práctica esta modalidad de trabajo ("*ojalá algún día consiga un curso de Diseño así puedo hacer esto con mis alumnos*").

Tal fue el entusiasmo de los alumnos que solicitaron un espacio de socialización en el que pudieran acercar estos aprendizajes a los demás compañeros de otros cursos y hacerlos partícipes de su experiencia. Esta inquietud dio origen, nada más ni nada menos, que a la propuesta de un Congreso organizado por el departamento, que se llevaría a cabo el año siguiente, en el que estos "casi" docentes presentarían una ponencia describiendo la metodología de trabajo ABP aplicada a la enseñanza de la programación.

De esta manera queremos poner en evidencia la importancia no sólo de enseñar una disciplina, en este caso la programación, sino de implicar profundamente al alumno del profesorado en su propio proceso de aprendizaje, de modo de potenciar su capacidad docente a la hora de ejercer su profesión. Asimismo, nos interesa puntualizar la necesidad de considerar la didáctica como una herramienta indispensable no solamente en los niveles primario y medio, sino -y sobre todo- en el nivel superior y específicamente en la formación docente, ya que es un pilar irreemplazable para evitar replicar modelos tradicionales que no se aplican a nuestra disciplina.

Poco ayudaría a un futuro profesor de nivel medio abundar en los conocimientos sobre teorías de lenguajes de programación o bases de datos -un conocimiento disciplinar que

nada tendrá que ver con su futuro laboral- y en contrapartida, dejar de lado la didáctica de la programación; es indispensable vincularla directamente con los contenidos de las cátedras. En este sentido no dudamos en que lo relevante está puesto en la forma en que se enseñan estos contenidos dentro de los profesorados, haciendo hincapié en la elaboración de una estrategia que permita a la vez, enseñar los conceptos disciplinares pero también la didáctica de su transmisión. Y no hay mejor manera de hacerlo que poner en juego esa didáctica en la propia cátedra.

En suma, con todo lo expuesto sostenemos que no pueden perseguirse los mismos objetivos al formar a un técnico informático que al formar a un docente de informática. Como dijimos en un principio, formar un docente de programación no solo implica que debe aprender cómo se resuelve un problema informático, sino que debe apuntar fundamentalmente, a la comprensión de los procesos que involucran tal aprendizaje. Para ello se hace necesario implementar estrategias que hagan explícita la manera en que se llega a esta comprensión, de modo de asegurar los mecanismos que luego se pondrán en juego a la hora de su posterior trasposición didáctica: sólo así podremos comenzar a darle estructura a una verdadera didáctica específica de la programación.

Bibliografía

Aiello, B., Iriarte, L., & Sassi, V. (2011). “La narración de la biografía escolar como recurso formativo”. *VIII Encuentro de Cátedras de Pedagogía de Universidades Nacionales*. La Plata: Memoria Académica – FaHCE.

Alliaud, A. (2004). “La experiencia escolar de maestros ‘inexpertos’. Biografías, trayectorias y práctica profesional”. En *Revista Iberoamericana de Educación* Nro. 34.

Alliaud, A. (2007). *La biografía escolar en el desempeño de los docentes*. Victoria: Universidad de San Andrés.

Brubacher, J., Case, C., & Reagan, T. (2000). *Cómo ser un docente reflexivo. La construcción de una cultura de la indagación en las escuelas*. Barcelona: Gedisa.

Davini, M. C. (2011). *Métodos de enseñanza*. Buenos Aires: Santillana.

Del Valle, Á., & Escribano, A. (2018). *El Aprendizaje Basado en Problemas: Una propuesta metodológica en Educación Superior*. Madrid: Narcea.

Edelstein, G. (2011). *Formar y formarse en la enseñanza*. Buenos Aires: Paidós.

Feldman, D. (2010). *Didáctica General*. Buenos Aires: MEN.

Maggio, M. (2018). *Reinventar la clase en la universidad*. Buenos Aires: Paidós.

Stasiejko, H., Pelayo Valente, J., & Krauth, K. E. (2012). Stasiejko, H., Pelayo Valente, J., & Krauth, K. E. (2012). *Concepciones de los docentes acerca del estudio en los ingresantes universitarios*. Buenos Aires: UBACyT.

Zeichner, K. (1993). *El maestro como profesional reflexivo*. Madrid: Morat.

e-Book | Actas

ISBN 978-950-33-1600-9

UNC FAMAF ffyh UNRC