

Revisión de propuestas para la enseñanza de la programación

Gustavo ASTUDILLO, Silvia BAST, Darío SEGOVIA & Leandro CASTRO

Departamento de Matemática/FCEyN/UNLPam
astudillo@exactas.unlpam.edu.ar, silviabast@exactas.unlpam.edu.ar,
dariosegovia2000@gmail.com, leajcastro@gmail.com

Resumen

Actualmente se pueden encontrar innumerables propuestas didácticas para que estudiantes de distintos niveles educativos aprendan a programar. En busca de identificar y categorizar estas propuestas se lleva adelante una revisión sistemática que permitirá responder las preguntas ¿Cuáles son esas propuestas a nivel nacional? ¿Cuál es la propuesta didáctica que las sustenta? ¿Qué herramientas utilizan? ¿Qué impacto ha tenido su implementación?

Habiendo revisado más de 80 trabajos, se presentan en este artículo los primeros resultados obtenidos y una primera clasificación de las propuestas.

Palabras clave: propuestas de enseñanza, programación, revisión sistemática

Introducción

El software está presente en la vida cotidiana de las personas y requiere, también, de personas que lo diseñen y construyan. Esto ha derivado en una tendencia mundial que promueve el aprendizaje de la programación, incluso desde los inicios de la escolarización. “En este escenario, la escuela, encargada de alfabetizar a la ciudadanía en saberes socialmente válidos, debe acercar a los estudiantes a la comprensión del mundo digital que nos rodea” (Busaniche, 2011; Echeveste & Martínez, 2016, p. 35).

Sin embargo, como afirman de la Fuente et al. (2018) “Enseñar y que los estudiantes aprendan a programar presenta diferentes dificultades y posibilidades de acuerdo a la edad y singularidad del aprendiz. Además del desafío de diseñar soluciones algorítmicas para resolver problemas específicos, el estudiante debe desarrollar competencias para expresarlas en un lenguaje en particular” (p. 1031).

Según Martínez & Echeveste (2015) las escuelas incorporan la computadora desde un enfoque predominantemente utilitario, son pocas las que estarían enseñando conceptos de computación relacionados con el pensamiento computacional y la programación.

Sin embargo, en Argentina, desde el 2018 se encuentran aprobados los “Núcleos de Aprendizajes Prioritarios de Educación Digital, Programación y Robótica” (Consejo Federal de Educación, 2018). Con esto se apunta a “facilitar la integración del acceso y dominio de las tecnologías de la información y la comunicación en los contenidos curriculares indispensables para la inclusión en la sociedad digital” (p. 1). Cada jurisdicción debe arbitrar los medios para su implementación.

Ante este cuadro de situación, surge una de las preguntas de investigación del proyecto ¿Cuáles son las propuestas didácticas que se están desarrollando en nuestro país para la enseñanza de la programación? y la presente

revisión sistemática para dar respuesta a la misma.

Metodología

La revisión sistemática, se realizó con base en la metodología propuesta por Kitchenham (2004), en la misma se propone: (a) definir preguntas de investigación, (b) trazar una estrategia de búsqueda (dónde buscar, con qué palabras claves), (c) establecer criterios de inclusión y exclusión, que serán aplicados, tanto para la selección inicial, como para la selección final.

(a) Preguntas de investigación

PI1 ¿Cuál es la estrategia propuesta para la enseñanza de la programación?

PI2 ¿A qué nivel educativo está dirigida la propuesta?

PI3 ¿Qué aprendizajes/habilidades se intentan desarrollar en los estudiantes?

PI4 ¿Cuál es la secuencia didáctica?

PI5 ¿Cuál es el enfoque pedagógico que sustenta la propuesta?

PI6 ¿Cómo se evalúan los aprendizajes?

PI5 ¿Qué recursos de software y hardware se utilizan para implementar las propuestas?

PI6 ¿Qué resultados se obtuvieron en la implementación de las propuestas didácticas?

(b) Estrategia de búsqueda

En esta primera etapa se intenta identificar las propuestas que se desarrollaron (y se están desarrollando) en nuestro país (Argentina). Se definió un ámbito de búsqueda a nivel nacional. Inicialmente, se comenzó por Jornadas/Congresos/Revistas con referato donde se aborden las temáticas de la Enseñanza de las Ciencias de la Computación. Se seleccionaron los Congresos/Jornadas WICC, CACIC, TE&ET, JADiPro y CICCSI por tener reconocimiento por la comunidad científica nacional. Las revistas seleccionadas fueron TE&ET y VEC. Luego de esta búsqueda inicial, se diversificó la misma en función de los hallazgos realizados en los artículos seleccionados (búsqueda en referencias bibliográfica y por autores).

La localización de artículos se realizó sobre las actas publicadas el periodo 2017 al 2018 inclusive. Luego, para los trabajos seleccionados en particular, se amplió el período hasta 2015. La búsqueda, dentro de las actas, se definió a través de palabras claves en idioma español tales como: programación, enseñanza, aprendizaje, pensamiento computacional y combinación de éstas.

Asimismo, se realizaron búsquedas por autor para los trabajos seleccionados y los pósters de JADiPro.

(c) Criterios de inclusión/exclusión

Para la inclusión de artículos se utilizaron los siguientes criterios:

CI1. Artículos cuyas propuestas se diseñan/implementan para Argentina.

CI2. Artículos accesibles de texto completo.

CI3. Artículos que abordan estrategias para la enseñanza de la programación en nivel secundario y/o universitarios.

Para la exclusión se utilizaron los criterios:

CE1. Referencias con idioma distinto al español o inglés

CE2. Artículos que a los que no se tuvo acceso al texto completo.

CE3. Artículos publicados en congresos/revistas sin referato.

CE5. Artículos que no responden a las preguntas de investigación.

Proceso preliminar

Se inició con la revisión de artículos que contuviesen las cadenas de búsqueda en título, palabras claves o resumen. Leído el resumen, se aplicaron los criterios de inclusión y exclusión.

Al ser localizadas referencias de interés, se obtenían las mismas y se aplicaban nuevamente los criterios de inclusión y exclusión.

Proceso de selección final

Terminada la fase preliminar, se comenzó con la lectura completa de los artículos en busca de responder las preguntas de investigación. Luego de estas lecturas, se volvieron a aplicar los criterios de inclusión y exclusión.

Marco teórico

La enseñanza de la programación trasciende el hecho de enseñar los tipos de estructuras y su utilidad, implica que el estudiante aprenda a analizar un problema, proponga uno o varios posibles diseños de solución, seleccione el más adecuado y luego pruebe que la solución a la que arribó es la correcta. En síntesis, que el estudiante aprenda a resolver problemas y expresar la resolución como una secuencia de instrucciones.

Se trata de un objetivo sumamente complejo, que implica abstracción, refinamiento, modularidad, entre otros. Compañ-Rosique et al. (2015) afirman “Para cualquier persona diseñar la solución a un problema requiere de un esfuerzo importante de abstracción, aún más si tiene que expresarla en forma de un algoritmo” (p.12). En este sentido Zúñiga et al. (2014) afirman que “la verdadera dificultad no reside en expresar la solución del problema en términos de instrucciones elementales, sino en el desarrollo del pensamiento computacional involucrado para la resolución del problema propiamente dicho”.

Se propone entonces, que debe desarrollarse en los estudiantes el pensamiento computacional, definido por Jeannette Wing (2006) como “El pensamiento computacional consiste en la resolución de problemas, el diseño de los sistemas, y la comprensión de la conducta humana haciendo uso de los conceptos fundamentales de la informática”.

Esta habilidad será de suma utilidad no sólo a quienes deseen dedicarse a profesiones relacionadas con la programación de software, sino a resolver problemas en otros aspectos de sus vidas.

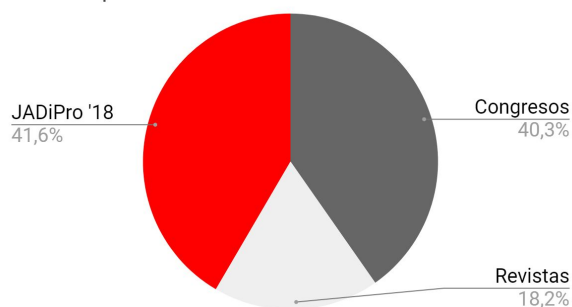
De la bibliografía analizada durante el relevamiento de propuestas, surge con claridad que una de las competencias claves a

desarrollar para el aprendizaje de la programación es el pensamiento computacional.

Resultados

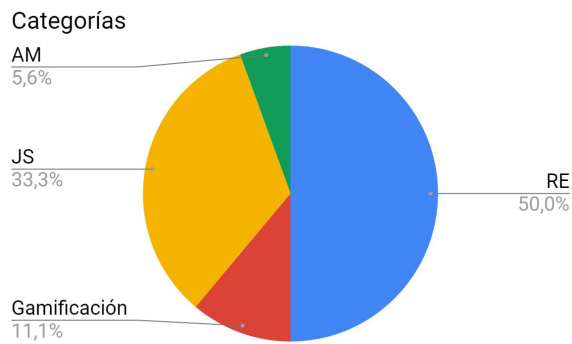
En el proceso de revisión se analizaron, al momento de escribir este artículo, 82 trabajos entre artículos y pósters (ver Gráfica 1).

Fuentes para la revisión sistemática



Gráfica 1. Distribución de las fuentes

Del análisis surgen, en principio, cuatro categorías: gamificación, juegos serios, robótica educativa y aprendizaje móvil, siendo la robótica educativa la que más producción presenta (ver Gráfica 2). En la primera, agrupamos aquellas que ludifican la propuesta e incluyen aspectos de los videojuegos en la misma. Dentro de juegos serios, se agrupan las que hacen uso de este tipo de juegos como eje central de la propuesta. En tercer lugar, las que usan la robótica educativa para llevar adelante las actividades; y finalmente, aquellas que incorporan los celulares en el proceso de enseñanza de la programación, las categorizamos como *aprendizaje móvil*. Cabe aclarar, que cada propuesta es incluida en la categoría en función de su eje central, dado que, en muchos casos hacen uso de recursos o estrategias que pertenecen a más de una categoría.



AM: Aprendizaje móvil, JS: Juegos serios, RE: Robótica educativa

Gráfica 2. Distribución por categorías

La Gráfica 2 representa el 66% de los artículos revisados, el resto se enfocan en didáctica de la programación (27%) y en otras área del conocimiento (7%).

Algunas propuestas

A continuación, por cuestiones de espacio, se presentan algunas de las propuestas identificadas y se exponen clasificadas según las categorías antes descriptas.

Gamificación

- Competir+Motivar+Hornero = aprender programación (Fracchia, Kogan, & Amaro, 2016). Esta propuesta de la Universidad Nacional del Comahue, emplea como estrategia el desarrollo de torneos de programación en los que “buscan potenciar la creatividad, la imaginación, el trabajo en equipo y el desarrollo del pensamiento computacional” (p. 19). En la propuesta los autores desarrollaron un juez en línea denominado HORNERO¹, herramienta web en la que desarrollan los torneos de programación. HORNERO permite trabajar con los lenguajes: java, python, php, c, c++, pascal, javascript, c#, ciao-prolog, perl, bash, lisp y ruby. Un torneo consiste en resolver correctamente todos los problemas que se presentan (10

por defecto). Cada solución “se ejecuta en la máquina del programador, recibiendo los parámetros y enviando la respuesta desde y hasta el servidor. Cada equipo se entera al instante si la solución enviada es correcta, de manera que puede corregirla y re-enviarla tantas veces como sea necesario. Durante el desarrollo de la competencia se visualiza el avance del tiempo y una tabla de resultados le permitirá conocer el número de problemas resueltos correctamente, el puntaje y el tiempo empleado en la resolución”. La propuesta está destinada, principalmente, al nivel medio (edades comprendidas entre 16 y 18 años) y se implementa a través de proyectos de extensión.

- Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación (Astudillo, Bast, & Willging, 2016; Willging et al., 2017). En esta propuesta, de la Universidad Nacional de La Pampa, los autores aplican estrategias de gamificación sobre un entorno Moodle. La secuencia, organizada por niveles, inicia con el uso de *lightbot*² para el aprendizaje de las nociones de secuencia, luego, ya en el nivel 2, se enfocan en las estructuras de control a través del uso de *code.org*³. En los niveles 3 y 4, se valen de *Scratch*⁴ para el abordaje de variables (simples y estructuradas) y de una introducción a la resolución de problemas. Esto a partir de la construcción, en Scratch, de un juego que se va desarrollando de forma iterativa acompañado/apoyado por el razonamiento experto. La propuesta está destinada a ingresantes a las carreras de Profesorado en Computación y Profesorado en Matemática, y es implementada desde 2010, como un taller en la Ambientación Universitaria.

¹ Disponible en: <http://hornero.fi.uncoma.edu.ar/>

² Disponible en: <http://lightbot.com/>

³ Disponible en: <http://code.org>

⁴ Disponible en: <https://scratch.mit.edu/>

- Plataforma gamificada que permite la resolución de desafíos en lenguaje C en el ámbito universitario (Vera et al., 2018, 2017). La propuesta, desde la Universidad Nacional de La Matanza, hace uso de la plataforma web gamificada “Programados” que “presenta los ejercicios de una forma distinta por medio de desafíos y permite la corrección automática de los mismos, tanto desde el punto de vista de la codificación como a través de pruebas lógicas”. Los estudiantes reciben puntos por cada solución e insignias por sus progresos, y se establecen *rankings*. Se implementa desde 2016 como una actividad extracurricular de la cátedra Elementos de Programación. Según el autor “Gamificar la plataforma ha sido parte del éxito dado que contribuye a la motivación de los alumnos, el interés por auto-superarse, la competencia entre cursos por alcanzar mejores puestos.” (Vera et al., 2018, p. 1100).
- Estrategias para la motivación e integración en el primer año universitario: Innovación mediante resolución de problemas y trabajo colaborativo (Ambort, Daniel & Castellaro, Marta, 2016). La propuesta de la Universidad Tecnológica Nacional, Regional Santa Fe, focaliza en la resolución de problemas, el aprendizaje basado en problemas y colaborativo. Incluye un conjunto de actividades y secuencias didácticas, que hacen uso de sitios juez en línea, campus virtual, problemas extra curriculares trabajados en equipo, participación en competencias, entre otras y además está basada en la resolución de problemas de ingeniería. Se muestran resultados alentadores de la aplicación de la misma.

Universidad Nacional de Quilmes (UNQ), López y sus colegas, proponen una metodología para el aprendizaje de nociones de programación basado en elementos concretos y enfocada en la abstracción, la cual “se refuerza mediante el uso intensivo de procedimientos y mediante la representación de elementos abstractos y su visualización.” (p. 7). La secuencia didáctica se sustenta en el aprendizaje basado en la indagación donde “los conceptos están ordenados en un nivel creciente de abstracción (por ejemplo, repetición simple antes de la repetición condicional [...])”. La plataforma gamificada presenta los ejercicios en forma de desafíos y permite la corrección automática de los mismos. Para su implementación hace uso de una herramienta del mismo nombre⁵ en la que los estudiantes hacen uso de un lenguaje textual o por bloques para resolver problemas planteados a partir de mover objetos dispuestos en una grilla. La propuesta es aplicada desde 2008 para introducir a la programación de computadoras a ingresantes de la UNQ, así como en escuelas del nivel medio. También es base de Program.AR⁶ y de Mumuki⁷.

- Encendiendo las luces del conocimiento con lightbot 1.0 (González, Paparoni, & Vallejos, 2017). Los autores presentan “una secuencia didáctica con Lightbot 1.0 elaborada para alumnos que cursan el primer año del secundario, con el principal objetivo de [...] permitir al alumno la construcción de significados de la programación, desarrollando a la vez habilidades del pensamiento computacional.” (p. 29). En esta propuesta, desarrollada en el Instituto Superior “San José” I-27-Corrientes, los

Juegos serios

- El método Gobstones (López, Ciolek, Arévalo, & Pari, 2017). Desde la

⁵ Disponible en:

<https://gobstones.github.io/gobstones-sr/>

⁶ Disponible en: <http://program.ar/gobstones-unq/>

⁷ Disponible en: <https://mumuki.io/home/>

autores definen tres fases: la primera, se basa en la resolución de los problemas planteados por el juego serio *lightbot* y la discusión de soluciones en clase junto al docente; y la segunda y tercera, con computación desconectada, usando papel y lápiz. Si bien se presentan algunos ejemplos de actividades, no se ofrecen datos sobre la implementación de la propuesta.

- Competición de Programación en la escuela secundaria. RITA en RED (Aybar Rosales, V., Queiruga, C. A., Kimura, I., Brown Bartneche, M., & Gómez, S., 2015), acción de extensión de la Universidad Nacional de La Plata, que focaliza en el aprendizaje de la programación en escuelas secundarias a través de RITA (*Remote Inventor to Teach Algorithms*) con un enfoque lúdico que incorpora estrategias de combate de robots virtuales en un campo de batalla a través de la programación con bloques. RITA ha sido diseñado con un doble propósito: i) promover el desarrollo de competencias de programación en bloques en el aula de la escuela, ii) favorecer la tarea docente a través del desarrollo de propuestas áulicas relacionadas a la enseñanza de programación.
- Pensamiento Computacional mediante Programación por Bloques: intervención didáctica usando Pilas Bloques. (Ahumada, Rivas, Contreras, Miranda, & Poliche, 2018). Este equipo de trabajo de la Universidad Nacional de Catamarca, evaluó el rendimiento académico de los estudiantes la asignatura Programación I, que es la que mayor cantidad de cursadas requiere, y afirma que tal situación se debe a la dificultad de los alumnos para desarrollar las capacidades básicas del Pensamiento Computacional y de los fundamentos de la Programación. Para revertir esta realidad presentan prácticas de enseñanza alternativas incorporando herramientas de programación por

bloques para facilitar y motivar el aprendizaje inicial de la programación.

Aprendizaje móvil

- Dispositivos Móviles como Soporte para el Aprendizaje Colaborativo de Programación en el Nivel Universitario Inicial (Reyes et al., 2017). Este proyecto, de la Universidad Nacional de Salta, toma como base los siguientes pilares: la Universidad como generadora de nuevos conocimientos, el software libre en las universidades nacionales y el aprendizaje móvil en contextos colaborativos como estrategias para mejorar los índices de ingreso y permanencia. Actualmente, se desarrolla la cuarta fase del proyecto con dos talleres colaborativos. El primero, destinado a estudiantes de primer año, en el que se lleva a cabo el redictado de la asignatura Elementos de Computación; y el segundo, destinado a ingresantes, cuyo objetivo es la motivación de los estudiantes en la resolución de problemas computacionales, a través del uso de un contexto lúdico y haciendo uso de aplicaciones de celulares.

Robótica Educativa

- FrankLab/Frankestito (de la Fuente et al., 2018; Picucci, Rodríguez, Zurita, & Cecchi, 2018; Zurita et al., 2017). Los autores, de la Universidad Nacional del Comahue, han desarrollado “una plataforma Web de robótica educativa, FrankLab, diseñada para favorecer el desarrollo de los procesos de enseñanza y de aprendizaje de conceptos y prácticas en el área Algoritmos y Lenguajes de Programación” (de la Fuente et al., 2018, p. 1030). La plataforma, destinada tanto a primaria como secundaria, “soporta Programación Tangible, Programación Basada en Bloques y Programación basada en texto”. La propuesta hace uso de robots físicos denominados Frankestito y N6-Frankestito (diseñados

y contruidos como hardware y software libre en la Facultad de Informática). Se encuentran desarrollando el primer prototipo. Sin embargo, los robots se han utilizado en actividades de divulgación científica e investigación en nivel medio.

- Diseño y programación de objetos interactivos digitales (Bordignon & Iglesias, 2015, 2016). Esta propuesta, de la Universidad Pedagógica Nacional, se basa en el “diseño, desarrollo y programación de objetos interactivos digitales [...] para desarrollar una propuesta educativa de introducción al pensamiento computacional” (p. 52). La misma se implementó a través de talleres basados en proyectos, donde a partir de una propuesta exploratoria, se introduce a estudiantes y docentes en el uso de placas Arduino, la programación y la electrónica básica. Los resultados fueron favorables, ya que “tanto estudiantes como profesores pudieron aprender juntos” y “los estudiantes pudieron desplegar de manera apropiada una capacidad de crear con tecnología digital tangible.” (Bordignon & Iglesias, 2016, p. 55).
- Uzi: máquina virtual sobre plataforma Arduino para robótica educativa (Morán, Zabala, Teragni, & Blanco, 2018). El proyecto, de la Universidad Abierta Interamericana, tiene como objetivo el desarrollo de una máquina virtual para Arduino que ejecute un set de instrucciones definido especialmente para facilitar la enseñanza de robótica en las escuelas. También buscan la Integración con *Physical Etoys*⁸. Actualmente, cuentan con un set de instrucciones (sintaxis inspirada en C) y un prototipo de la máquina virtual. Asimismo, se encuentran desarrollando la propuesta de intervención didáctica y los instrumentos de recolección de datos.

- Juegos interactivos en ARDUINO y Java, para motivar y despertar el interés en Informática (Belcastro et al., 2017). Esta propuesta, de la Universidad Nacional de La Plata, focaliza en “apoyar a los estudiantes, y favorecer el autoaprendizaje, la autonomía y la creatividad. Introduce recursos de la Web 2.0, características del aprendizaje significativo, herramientas de trabajo colaborativo, que se emplearon dentro del proyecto y apoyan el carácter adaptativo del trabajo del desarrollador de software.” (p. 1256). La propuesta se implementó a través del evento de transferencia “juego interactivo en java. Introducción al sistema binario” donde alumnos “interactuaron con un juego simple, desarrollado en java [y en Arduino programaron un] teclado binario capaz de interpretar pulsos electrónicos, y mostrar en pantalla un carácter codificado en ASCII” (p. 1259). Tuvieron resultados alentadores de acuerdo a las encuestas.
- Programando con el robot educativo en la UNNOBA (Russo, Ramón, Lencina, Adó, & Iglesias, 2017). Los autores, de la Universidad Nacional del Noroeste de la Provincia de Buenos Aires (UNNOBA), han diseñado una interfaz para un robot educativo programable (REP) basado en Arduino. El objetivo es “analizar y desarrollar herramientas y estrategias didáctico-pedagógicas que permitan el diseño y la evaluación de actividades para la inclusión de las TICs en las actividades áulicas” (p. 355). La propuesta está destinada a ingresantes de la UNNOBA y hacen uso del lenguaje Python para controlar el REP. Según se indica en el artículo se está en proceso de desarrollo de la propuesta.

Conclusiones

En este artículo se presentan los primeros resultados de una revisión sistemática que se lleva adelante con el objetivo de identificar y

⁸ Disponible en:
<http://tecnodacta.com.ar/gira/projects/physical-etoys/>

clasificar las propuestas de enseñanza de la programación a nivel nacional (Argentina).

A partir del relevamiento, se han podido identificar una ingente cantidad de propuestas y definir una clasificación, preliminar, en cuatro categorías: gamificación, juegos serios, aprendizaje móvil y robótica educativa; siendo esta última la que mayor cantidad de propuestas registra.

Como trabajos futuros, queda el análisis del resto de los artículos hallados y la ampliación de la búsqueda con el fin de construir un mapa de propuestas, a nivel país, lo más ajustado a la realidad posible.

Asimismo, queda por delante, una segunda clasificación de las propuestas en función del sustento pedagógico de las mismas, y a partir de allí, retroalimentar el diseño de una secuencia didáctica, para el aprendizaje de nociones básicas de programación, que se encuentra en desarrollo basada en el uso de robótica educativa.

Bibliografía

- Ahumada, H. C., Rivas, D. A., Contreras, N., Miranda, M. del V., & Poliche, M. V. (2018). Pensamiento computacional mediante programación por bloques: intervención didáctica usando pilas bloques. *XIII Congreso de Tecnología en Educación y Educación en Tecnología (Posadas, 2018)*.
- Ambort, Daniel, & Castellaro, Marta. (2016). Estrategias para la motivación e integración en el primer año universitario: Innovación mediante resolución de problemas y trabajo colaborativo. *Virtualidad, Educación y Ciencia*, 7(12), 85–98.
- Astudillo, G. J., Bast, S. G., & Willging, P. A. (2016). Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación. *Virtualidad, Educación y Ciencia*, 7(12), 125–142.
- Aybar Rosales, V., Queiruga, C. A., Kimura, I., Brown Bartneche, M., & Gómez, S. (2015). Enseñando a programar con RITA en escuelas secundarias. *Memorias del XXI Congreso Argentino de Ciencias de la Computación (Junín, 2015)*, 11. <http://sedici.unlp.edu.ar/handle/10915/50028>
- Belcastro, Á., Alanes, R. D., Quiroga, M., Giménez, J., Santana, S., Dibez, P., & Bertone, R. A. (2017, abril). *Juegos interactivos en ARDUINO y Java, para motivar y despertar el interés en Informática*. Presentado en XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires). Recuperado de <http://hdl.handle.net/10915/62922>
- Bordignon, F. R. A., & Iglesias, A. A. (2015). *Diseño y construcción de objetos interactivos digitales*. Recuperado de <http://hdl.handle.net/10915/50448>
- Bordignon, F. R. A., & Iglesias, A. A. (2016). Más allá de las pantallas: experiencias en diseño y programación de objetos interactivos digitales. *Virtualidad, Educación y Ciencia*, 7(12), 49-58.
- Busaniche, B. (2011). Analfabetización informática o ¿ Por qué los programas privativos fomentan la analfabetización? En *Construcción Colaborativa del Conocimiento* (pp. 177-187). Universidad Nacional Autónoma de México. Instituto de Investigaciones Económicas.
- Compañ-Rosique, P., Satorre-Cuerda, R., Llorens-Largo, F., & Molina-Carmona, R. (2015). Enseñando a programar: un camino directo para desarrollar el pensamiento computacional. *Revista de Educación a Distancia*, (46). Recuperado de <https://revistas.um.es/red/article/view/240191>
- Consejo Federal de Educación. Nucleos de Aprendizajes Prioritarios (NAP) de Educación Digital, Programación y Robotica. , 343/18 CFE § (2018).
- de la Fuente, J., Picucci, M., Bonet Peinado, D., Zurita, R., Parra, G., Rodríguez, J., & Cecchi, L. (2018). Construyendo franklab: una plataforma web de robótica educativa.

- XXIV Congreso Argentino de Ciencias de la Computación*, 1030-1039. Recuperado de <http://sedici.unlp.edu.ar/handle/10915/73601>
- Echeveste, M. E., & Martínez, M. C. (2016). Desafíos en la enseñanza de Ciencias de la Computación. *Virtualidad, Educación y Ciencia*, 7(12), 34-48.
- Fracchia, C. C., Kogan, P., & Amaro, S. (2016). Competir+ motivar+ hornero= aprender programación. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, (18), 19-29.
- González, J., Papanoni, V., & Vallejos, L. (2017). Encendiendo las luces del conocimiento con lightbot 1.0.: la construcción del conocimiento en la clase de computación. *XII Congreso de Tecnología en Educación y Educación en Tecnología*, 29-38. Recuperado de <http://sedici.unlp.edu.ar/handle/10915/63366>
- Kitchenham, B. (2004). *Procedures for performing systematic reviews* (Technical Report N.º TR/SE-0401; p. 27). Recuperado de Keele University website: http://tests-zingarelli.googlecode.com/svn-history/r336/trunk/2-Disciplinas/MethodPesquisa/kitchenham_2004.pdf
- López, P. E. M., Ciolek, D., Arévalo, G., & Pari, D. (2017). The GOBSTONES method for teaching computer programming. *2017 XLIII Latin American Computer Conference (CLEI)*, 1-9. IEEE.
- Martínez, M. C., & Echeveste, M. E. (2015). Representaciones de estudiantes de primaria y secundaria sobre las Ciencias de la Computación y su oficio. *Revista de Educación a Distancia*, (46). Recuperado de <https://revistas.um.es/red/article/view/240241>
- Morán, R., Zabala, G., Teragni, M., & Blanco, S. (2018). *Uzi: máquina virtual sobre plataforma Arduino para robótica educativa: resultados de la primer etapa*. Presentado en XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste). Recuperado de <http://sedici.unlp.edu.ar/handle/10915/68687>
- Picucci, M., Rodríguez, J., Zurita, R., & Cecchi, L. (2018). *Entorno web para la programación del robot Frankestito*. Presentado en XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste). Recuperado de <http://hdl.handle.net/10915/67520>
- Reyes, C. J., Massé Palermo, M. L., Espinoza, C. N., Vargas, C., Ramírez, J., & Trenti, J. (2017). Dispositivos móviles como soporte para el aprendizaje colaborativo de programación en el nivel universitario inicial (resultados). *XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires)*.
- Russo, C. C., Ramón, H. D., Lencina, P., Adó, M., & Iglesias, P. (2017). Programando con el robot educativo en la UNNOBA. *XII Congreso de Tecnología en Educación y Educación en Tecnología*, 353-361. Recuperado de <http://hdl.handle.net/10915/63439>
- Vera, P. M., Moreno, E. J., Rodríguez, R. A., Vallés, F. E., Tamashiro, S., & Vogel, F. (2018). *Plataforma gamificada que permite la resolución de desafíos en leguaje C en el ámbito universitario*. Presentado en XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste). Recuperado de <http://sedici.unlp.edu.ar/handle/10915/68394>
- Vera, P. M., Moreno, E. J., Rodríguez, R. A., Vázquez, M. C., Vallés, F. E., & Cescon, J. G. (2017, abril). *Gamificación en el ámbito universitario*. 1230-1234. Recuperado de <http://hdl.handle.net/10915/62917>
- Willging, P., Astudillo, G. J., Bast, S. G., Occelli, M., Castro, L., & Distel, J. M. (2017). Educación con Tecnologías: la Gamificación Aplicada para el Aprendizaje de la Programación. *XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos*

Aires).

Wing, J. (2006). *Computational Thinking. It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.*

Communications presentado en ACM.

Recuperado de

<https://www.cs.cmu.edu/~15110-13/Wing06-ct.pdf>

Zúñiga, M. E., Rosas, M. V., Fernández, J., & Guerrero, R. A. (2014, octubre 15). *El desarrollo del pensamiento computacional para la resolución de problemas en la enseñanza inicial de la programación.*

Presentado en XVI Workshop de

Investigadores en Ciencias de la

Computación. Recuperado de

<http://hdl.handle.net/10915/41352>

Zurita, R., Fuente, J. de la, Bucarey, M.,

Bonet, D., Castillo, R. del, Grosso, G., ...

Rodríguez, J. (2017, junio). *Mejorando las posibilidades de aprender a programar,*

ampliación del robot educativo múltiple N6

Max a Frankestito. Presentado en XII

Congreso de Tecnología en Educación y

Educación en Tecnología (TE&ET, La

Matanza 2017). Recuperado de

<http://hdl.handle.net/10915/63450>