

CÓDIMO: desarrollo del pensamiento computacional en las escuelas

Graziani, Luciano Martín; Sanhueza, Matías Emanuel; Cayú, Gabriela Anahí

Docentes Tutores: Molinari, Enrique Pablo; Britos, Paola Verónica; Alasio, María Silvia

Universidad Nacional de Río Negro, Argentina

{lgraziani, msanhueza, gcayu}@unrn.edu.ar

{emolinari, pbritos, malasio}@unrn.edu.ar

Innovación en Centros Educativos y de Investigación

Resumen

El Pensamiento Computacional (de aquí en más, PC) debe ser integrado dentro de la currícula escolar de la manera más transparente posible, teniendo en cuenta que la computación es un área transversal a todas. Como dijo Jeannette Wing en el 2006, “tener habilidades del Pensamiento Computacional es un requisito indispensable para todas las personas”. Saber PC en la era digital es útil para tener un mejor desempeño en los distintos ámbitos de la vida. En este trabajo presentamos cómo fue nuestra experiencia en distintas actividades extracurriculares y qué nos llevó a plantearnos la necesidad de crear una aplicación para el desarrollo del PC con el objetivo de que cualquier docente de cualquier materia pueda utilizarla.

Palabras clave: educación, pensamiento computacional, programación, tics.

Introducción

Desde el año 2014, la Universidad Nacional de Río Negro ha realizado distintas actividades para llevar la programación a las escuelas primarias y secundarias de Viedma. Durante todas ellas, el objetivo principal fue contribuir a despertar vocaciones tempranas en carreras afines a las Tecnologías de la Información y la Comunicación (de aquí en más, TICs). La falta de información que poseen los jóvenes con respecto a las Ciencias de la Computación, tanto en el nivel primario como en el secundario, nos llevó a plantear la necesidad de informar y concientizar a la población escolar en este tema.

Despertando vocaciones: los inicios

En el primer semestre del año 2014 bajo el marco de la iniciativa Program.ar la Fundación Sadosky brindó una capacitación¹ —con sede en Viedma—, en la que participamos junto a profesores y otros alumnos de la Universidad Nacional de Río Negro y la Universidad Nacional de la Patagonia San Juan Bosco, con el objetivo de despertar vocaciones tempranas en carreras afines a las TICs.

Durante la capacitación concurrimos a una escuela secundaria donde cumplimos dos roles: fuimos estudiantes a la hora de aprender a enseñar y fuimos ayudantes cuando enseñábamos lo que aprendíamos. Todo en una clase y en un solo lugar. De a poco, cada uno de nosotros fue adoptando

¹ Más información en <https://goo.gl/VNISXg>.

una forma distinta de reflejar los conocimientos en el otro y comprendimos que aprender y enseñar son dos reactivos en la combustión de una misma llama donde cada uno alimenta al otro.

La metodología planteada por la Fundación Sadosky para enseñar en las escuelas fue a través del desarrollo de pequeños cursos sobre programación de animaciones y videojuegos. Para tal propósito, utilizamos la herramienta Alice² (una aplicación desarrollada por la Universidad de Carnegie Mellon), por la facilidad en *escribir* código mediante bloques en forma de rompecabezas y por su rápida respuesta visual. Alice provee personajes (ya sean animales o personas), escenarios, árboles, casas y otros elementos que captan la atención del *programador* y que permiten asociar el código de bloques con lo que ocurre en la animación.

Llevando la programación a las escuelas

Con la motivación alta y las herramientas listas, era nuestro turno para sembrar el interés en las carreras afines a las TICs. A lo largo del segundo semestre del 2014, contactamos a las escuelas para presentar nuestra propuesta y siempre recibimos respuestas favorables de todas ellas.

La falta de proyectores, la escasez de enchufes para conectar las netbooks o el espacio reducido fueron algunos desafíos que se nos presentaron a medida que avanzábamos con los cursos. A pesar de ello, pudimos desenvolvemos bien y cumplir con nuestro objetivo.

Otro desafío, ciertamente más particular, fue la relación que tuvimos con los alumnos. El enfoque de enseñanza que aplicamos nos permitió identificar algunos de los distintos tipos de estudiantes con quienes es posible encontrarse en un curso de esta índole.

De esto último, lo más importante fue poder conocer alumnos que adquirieron el conocimiento y lo transformaron en una idea original, por ejemplo, al realizar más de lo que la actividad planteaba. Además, observamos que tomando sus trabajos como ejemplo, incentivábamos al resto a hacer cosas similares. Así, siempre buscamos crear un clima de motivación, imaginación y creatividad.

El enfoque de enseñanza nombrado anteriormente consistió en una serie de pasos que se repetían a medida que se avanzaba en las actividades:

1. Presentar la actividad.
2. Explicar, a través del uso de Alice, los pasos iniciales y conceptos vistos en actividades pasadas.
3. Dar pie a la inquietud e incertidumbre de querer solucionar un problema relacionado con el concepto que se quiere enseñar.
4. Ofrecer un tiempo determinado para que los alumnos intenten solucionar el problema.
5. Presentar el concepto a través de la solución del problema, comparándola con algunas de las soluciones propuestas por los alumnos.
6. Destacar ejercicios de los alumnos y mostrarlos al resto de la clase para motivar e incentivar la imaginación del resto.

² Para más información, acceda a <http://www.alice.org/index.php>

¿Qué hace que un alumno opte no por carreras afines a las TICs?

Al inicio del 2015, la fundación DAR (Desarrollo Argentino) junto con MICROSOFT realizaron una convocatoria al público en general para participar del primer Tecnotour en la localidad de Viedma³. Este evento nos dio pie para formular el primer esbozo de lo que en un futuro llegaría a ser CÓDIMO.

Planteamos el problema de la ausencia de la enseñanza de la programación en las escuelas de Viedma, y mencionamos que el Consejo Federal de Educación, en la Resolución 263/15, establece: “Considerando [...] que hay abundante evidencia científica que indica que los niños/as y adolescentes que aprenden a Programar, mejoran su desempeño en otras áreas disciplinarias [...]. Por ello resuelve: Artículo 1º: Establecer que la enseñanza y el aprendizaje de la Programación es de importancia estratégica en el Sistema Educativo Nacional durante la escolaridad obligatoria, para fortalecer el desarrollo económico-social de la Nación”.

Por otro lado, también abordamos la falta de información en los jóvenes con respecto a las Ciencias de la Computación tanto a nivel primario como secundario. Principalmente, los alumnos no tienen buenas experiencias luego de cursar la materia relacionada con informática: la encuentran como algo tedioso, donde uno se dedica a trabajar con un procesador de texto o una planilla de cálculos, y donde se hacen actividades completamente ajenas al resto de las materias. En segundo lugar, aparece el temor de cursar una carrera “extremadamente difícil” con una base matemática fuerte, completamente superior al nivel con el que un alumno promedio se egresa de la secundaria. Finalmente, los temores y las malas experiencias ayudan a que los estudiantes opten por otras carreras.

Creemos que el sistema educativo no necesita más contenido sino, particularmente, entornos flexibles que permitan desarrollar las capacidades de autoaprendizaje, creatividad, autonomía, iniciativa y trabajo en equipo.

Primera solución

Como respuesta a estos problemas, planteamos el desarrollo de un pseudo-lenguaje en castellano para ser utilizado en actividades que forman parte de la currícula escolar. De esta manera, asegurábamos poder implementar una herramienta para asentar los conceptos que se enseñan en una materia a través de un lenguaje de programación en castellano y así poder romper la barrera idiomática de los lenguajes de programación.

Sin embargo, esta propuesta estaba lejos de poder ser implementada. Principalmente, porque requiere que el docente se capacite en programación, ya que por más que fuera un pseudo-lenguaje en castellano, sigue siendo programación, con las reglas que conlleva. A su vez, el alumno se enfrentaría a problemas básicos como: corregir errores de tipeo, escritura correcta del código y su lectura para encontrar un error que no sea de tipeo.

³ Más información en <https://goo.gl/GWus9t>.

CÓDIMO: CÓDigo, IMaginación y MOtivación

Para la segunda convocatoria del Tecnotour⁴, solucionamos esa problemática al abstraer por completo la escritura de código a través del uso de bloques. De esta manera, además de ser más sencillo para los docentes y los alumnos, sería más realista y simple de utilizar. A esta solución la llamamos CÓDIMO.

Además de la simplificación de la herramienta, la enmarcamos bajo el concepto de PC. Según Jeannette Wing, el PC *implica la resolución de problemas, el diseño de los sistemas y la comprensión del comportamiento humano haciendo uso de los conceptos fundamentales de la informática*. En ese mismo artículo continúa diciendo que *esas son habilidades útiles para todo el mundo, no sólo para los científicos de la computación* (Wing, 2006).

Luego de competir en las finales locales del Tecnotour, presentamos el trabajo al director de carrera con el fin de buscar apoyo docente para poder llevar a cabo la aplicación. A partir de esa iniciativa, integramos a un jefe de trabajos prácticos de materias de programación, y una profesora de Lengua que forma parte del equipo de la carrera.

Durante las primeras reuniones, definimos con mayor certeza las implicancias de CÓDIMO, sus metas, qué es y qué no es. A partir de ello, lo presentamos como proyecto de extensión, en la convocatoria realizada por la Universidad a fines del año 2015. El proyecto llamado “Imaginación y Motivación. Puntos de partida para la enseñanza de la Programación en las escuelas”⁵ fue aprobado a fines del primer semestre del siguiente año, con un año de duración. En él, pretendemos desarrollar un prototipo de CÓDIMO que contenga una actividad de matemática y otra de ciencias sociales para séptimo grado de la escuela primaria. Luego será puesto a prueba en una escuela de la ciudad de Viedma para observar de qué manera los alumnos reciben el programa.

Implicancias y futuro

CÓDIMO será una herramienta de apoyo para el docente con la cual pueda incentivar a sus alumnos a solucionar problemas sobre conceptos que ya fueron dados en la clase. Esto quiere decir que no intentará reemplazar los recursos que ya tiene un docente para enseñar, sino que brindará un refuerzo, una manera de fijar los conocimientos utilizando una herramienta digital que incentive a los alumnos y que al mismo tiempo sea accesible para cualquier docente.

CÓDIMO buscará, como meta final, que los alumnos puedan aplicar los conocimientos aprendidos en el aula a través actividades no repetitivas. Deberán aplicar los conceptos desde un punto de vista distinto con el fin de adoptar un poco de “Pensamiento Computacional”.

CÓDIMO será una aplicación web, para eliminar la necesidad de instalación, y consistirá en un repositorio de actividades que formen parte de la currícula escolar. Cada actividad contendrá un

⁴ Más información en <https://goo.gl/hZ7JaU>.

⁵ Aprobada por la [Resolución Rectoral 315/16 de la UNRN](#).

conjunto limitado de acciones a realizar y estará diseñada para que siempre tenga una representación animada de lo que está realizando el alumno.

CÓDIMO no será un nuevo lenguaje, y por ello no brindará la posibilidad de escribir código. Tampoco será una aplicación autocontenida, por lo que el uso de internet es necesario.

Evidencias científicas que demuestran los beneficios

Jesús Moreno León, en su artículo publicado en el sitio web programamos.es, “Evidencias científicas de los beneficios de aprender a programar desde infantil”, realiza un análisis de las evidencias que existen y que demuestran los beneficios de enseñar programación en edades tempranas.

Cuando se comenzó a introducir el lenguaje de programación Logo en la currícula de muchas escuelas de Estados Unidos, distintos grupos de investigadores comenzaron a estudiar qué otras cosas aprendían los alumnos, además de la programación, durante los cursos. Así, en el año 1986, el artículo “Effects of Logo and CAI environments on cognition and creativity” (Clements, 1986) explicaba en sus conclusiones que los niños que usaron Logo en educación infantil demostraron mayor capacidad de atención, más autonomía, y mostraban un mayor placer por el descubrimiento de nuevos conceptos.

[...] Por último, existen evidencias científicas que demuestran que el alumnado que aprende a programar en edades tempranas tiene menos estereotipos de género en relación con las carreras STEM - Ciencias, Tecnología, Ingeniería y Matemáticas- (Women and Minorities in Science, Technology, Engineering and Mathematics) y menos reticencias para continuar sus estudios y profesiones en estas disciplinas (Gender related to success in science and technology). (León, 2014).

Conclusión

Creemos que es indispensable que cualquier docente, sin importar su asignatura, tenga herramientas digitales con las que reforzar los conocimientos que los alumnos aprenden en la clase. Con el desarrollo de CÓDIMO queremos además que dicha herramienta sea atractiva para los alumnos y simple de usar para un docente. CÓDIMO no reemplaza otros recursos didácticos pero hace que, al final de cada ejercicio, todos los participantes se lleven un poco de Pensamiento Computacional.

Referencias

- Clements, D. H (1986). Effects of Logo and CAI environments on cognition and creativity. *Journal of Educational Psychology*, 78(4) 309-318.
- León Moreno, J. (2014). Evidencias científicas de los beneficios de aprender a programar desde infantil Programamos. [versión electrónica]. Recuperado de <http://goo.gl/TFQ1pK>. Vigente octubre 2016.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49:33–35. [versión electrónica]. Recuperado de <https://goo.gl/CHS4Yp>. Vigente octubre 2016.