

POSTÍTULO

ACTUALIZACIÓN ACADÉMICA EN EN EDUCACIÓN DIGITAL, PROGRAMACIÓN Y ROBÓTICA

Responsables: Cecilia Martínez | Cristian Zabala | Nicolás Wolovick | Eduardo Rodríguez

| Carolina Wayar | Marcos Gómez | Cristian Rojo | Leandro Vélez.

Autores: Gabriel Caiero | Cecilia Martínez | Mariana Pérez | Leandro Vélez | Cristian Zabala.

Coordinadores académicos: Gabriel Caiero | Mariana Pérez.



Secretaría de
Educación



Municipalidad
de Córdoba

ese proceso, ofrecemos oportunidades a nuestros estudiantes de construir escritos para diferentes fines. En tal sentido, enseñar a programar no se presenta necesariamente como un fin en sí mismo, sino como la mejor estrategia que conocemos para introducir conceptos de las Ciencias de la Computación de una manera significativa para los chicos y las chicas en edad escolar. Por aprendizaje significativo nos referimos a un proceso donde los conceptos se construyen y profundizan gradualmente y donde la tarea de programar y los conceptos necesarios para hacerlo tienen un sentido, un propósito para quien las hace. Con la salvedad que programación es solo una parte de todo lo que abarca las Ciencias de la Computación.

No obstante, contrario al aprendizaje de la lectura y escritura, la programación está sujeta a una variedad de lenguajes que poseen una sintaxis y semántica propia por lo tanto supone diferentes formas de abordar los problemas y de producir estructuras procesables en la computadora (Muraro, 2020. Intercambios informales).

Codificación: Refiere a la etapa de implementación de soluciones en un lenguaje de programación determinado. En términos de las operaciones mentales, aborda las operaciones expresivas o simbólicas en tanto representa una solución a un problema en un lenguaje que una computadora pueda comprender. Por contraste, la programación, aborda operaciones reflexivas (que incluyen análisis, comprensión), creativas (que incluyen transferir, desarrollar, predecir, imaginar) y prácticas (usar y aplicar herramientas para solucionar un problema). De hecho, para Papert (1987) programar es esencialmente crear con tecnología y construir con una computadora. Más allá de ofrecer instrucciones precisas, la construcción es representación de ideas y la posibilidad de expresar una forma de resolver problemas a través de un lenguaje de programación.

Para seguir con nuestra comparación con el aprendizaje de la lecto escritura, la codificación sería equivalente a escribir, pero desacoplado de la elaboración de un texto con sentido. Es decir, alguien puede escribir un texto dictado o copiado de otro, o es posible escribir respuestas a preguntas o consignas y con eso podríamos decir que esa persona está alfabetizada ya que es un indicador de alfabetización en la lecto escritura. Sin embargo, organizar ideas en un texto, elegir el tipo de texto adecuado para la función comunicativa que necesitamos, apropiarse de las reglas del tipo de escritura que ese texto requiere, necesita de otro nivel de apropiación de la lecto escritura que podríamos llamar pensamiento con en

lenguaje. Es importante poder entender esta diferencia porque, como veremos más adelante, a principio de los 80s se enseñaba "computación" a partir de copiar "código " de una revista o libro a una máquina. Si bien esto no estaba institucionalizado en las escuelas, muchos chicos y chicas que accedían a la programación lo hacían por estos medios. Esta práctica de enseñanza que se fue heredando y dejando huellas en algunas publicaciones actuales. Es posible ver algunos libros y programas educativos actuales que ofrecen ese formato. Como venimos mostrando, esta copia de código no necesariamente desarrolla el pensamiento computacional, lo que desarrolla el pensamiento computacional es pensar problemas computacionales. Sin embargo no toda práctica de copiar y pegar es considerada un mero aplicacionismo sin sentido, en la tesis de Echeveste (2020) copiar y pegar se complementa con saber buscar y saber preguntar, las cuales son prácticas que realizan los y las estudiantes cuando están programando y se ligan a las prácticas propias del pensamiento computacional. Incluso Brennan y Resnik (2012) lo consideran dentro de las prácticas del PC ya que permite "reusar y remezclar" como actividad de este pensamiento.

En este contexto, el término Pensamiento Computacional tiene un uso para la política educativa y otro técnico. Dentro del uso técnico se definen diferentes habilidades y saberes tales como abstraer, analizar y diseñar artefactos computacionales. El uso para la política educativa del término, adoptado mayormente por los países del Norte, es para diferenciar y criticar las formas de alfabetización digital que se han desarrollado tradicionalmente en la escuela con foco en el uso de software y las Tecnologías de la Información y la Comunicación.

La importancia de establecer definiciones claras sobre estos términos sirve para diferenciar las prácticas computacionales que la escuela ofrecía en ese momento (centrada en el uso del software) y la prácticas que se esperaba que la escuela pudiera ofrecer para aumentar el interés de los estudiantes en las carreras TIC y STEM y cerrar las brechas digitales entre diferentes segmentos de la población. De esta manera, se pudo establecer que la escuela no solo tiene que enseñar computación o TIC -términos ambiguos- sino especialmente pensamiento computacional y programación si quería abordar la brecha digital y la formación del ciudadano en un mundo digital.

En nuestro país, el desarrollo de estas precisiones conceptuales fue impulsada por referentes de la comunidad de software libre - entre otros actores-.. A partir del

año 2006 la referente de la organización Vía Libre, Beatriz Busaniche, publica una serie de artículos en donde establece que alfabetizar, según la UNESCO, es aprender un lenguaje para poder producir y crear a partir de él. Busaniche establece que la escuela solo enseñaba iconografía básica creando la ilusión de aprender computación, pero que no estaba ofreciendo lenguajes computacionales que permitieran a los jóvenes: acceder y modificar software disponible y crear software que permita resolver sus propios problemas. Denomina a este proceso como “Oscurantismo Informático” en tanto que se mantienen oculto al general de la población los modos de escritura de nuestro tiempo -tal como sucedió en la época medieval- que permiten desarrollar procesos digitales.

Tres motivos para enseñar a programar:

- Alfabetización digital para participar del mundo digital
- Aprender a programar puede potenciar otros tipos de pensamiento
- Aprender a programar es la mejor forma que conocemos de acceder a las Ciencias de la Computación

3. PROYECTOS INTERDISCIPLINARIOS DE EDUCACIÓN DIGITAL, PROGRAMACIÓN Y ROBÓTICA

3.1. Presentación

En esta clase reflexionamos sobre qué necesitamos conocer para orientar experiencias interdisciplinarias de enseñanza que incluyan el pensamiento computacional en nuestras escuelas. Por ello, avanzaremos sobre los principales componentes y criterios de una experiencia educativa interdisciplinaria.

La programación, como vimos, requiere ineludiblemente poner en juego procesos propios del pensamiento computacional. Trabajar en contextos donde se invita a los estudiantes a diseñar y programar su propio autómeta: videojuego, animación, robot, chat, sensores o lo que fuere; no solo involucra habilidades cognitivas del pensamiento computacional, sino también requiere habilidades cognitivas y competencias propias de otras áreas del conocimiento en tanto que la programación se desarrolla para aportar a la resolución de un problema. Los problemas ambientales, sociales, educativos, sanitarios, etc; tienen siempre diferentes dimensiones. Imaginemos una situación de contaminación de agua. Una dimensión es la normativa de las políticas públicas que designa qué tipos de agua se pueden tirar al río y qué tipos de agua deben ser tratadas. Otra dimensión es la formación del ciudadano. Otra la química, que nos sirve para identificar la contaminación propiamente dicha.

Incluir el pensamiento computacional en un problema o proyecto, significa que podamos pensar en el aporte que puede hacer el cómputo, el procesamiento de datos o la automatización de tareas por parte de una máquina, a este problema.

Por ejemplo, automatizar válvulas de apertura o cerramiento de caños según niveles de vertederos, programar sensores para que den aviso cuando la composición química del agua es peligrosa, etc. Es muy importante tener claridad que usar una computadora para modificar imágenes sobre el tema (presentaciones en power point, videos, gráficos, etc) no requiere del pensamiento computacional. El proyecto interdisciplinario incluye el pensamiento computacional cuando se piensa en automatizar una tarea, o procesar y transformar datos.

Para saber más: Las relaciones entre experiencias de enseñanza de la programación y la matemática son quizás las más evidentes, no obstante, los proyectos de programación también pueden vincularse a la música, el arte, las

ciencias sociales, las ciencias naturales, etc, como vimos en la actividad 1 de esta clase. La página de Scratch presenta varios proyectos interdisciplinarios que incluyen la programación. En el artículo “Estado del arte sobre experiencias de enseñanza de programación a niños y jóvenes para el mejoramiento de las competencias matemáticas en primaria” dos investigadores colombianos, Palmas Suárez y Sarmiento Porras (2015), revisan diferentes experiencias de enseñanza de la programación y su vinculación con el desarrollo y mejoramiento de competencias matemáticas.

Si bien la programación ha ingresado, desde hace más de 20 años, a muchas escuelas, también hemos observado que su enseñanza no siempre refleja el espíritu constructorista que proponen Papert y Resnick. En ocasiones, hemos visto que se copia el código en una pizarra y los alumnos deben re-escribirlo en su computadora, o bien deben seguir una serie de instrucciones para armar un robot o programar. Estas actividades no estimulan el pensamiento computacional porque se centran en capacidades de bajo orden como seguir instrucciones o copiar y pegar. Los proyectos interdisciplinarios, en cambio, requieren de pensar en problemas donde se involucre también el cómputo. El pensamiento computacional necesita tener un problema computacional con el que el estudiante se enfrenta para poder resolverlo. Es por ello que pensar en diseñar experiencias de enseñanza que incluyan la programación es tan importante como transmitir los conceptos de ciencias de la computación. A continuación, recuperamos a algunos autores para ofrecer elementos y criterios para diseñar una clase de programación que desarrolle el pensamiento computacional.

3.2. **Contenidos**

Criterios para diseñar proyectos de enseñanza que incluyan la educación digital, programación y robótica. Enfoque del constructorismo de Papert. Aprendizaje por indagación y resolución de problemas y desafíos. Momentos para la elaboración de proyectos. Las dimensiones de la planificación educativa y las variables a considerar al momento de la introducción de tecnología en el aula.

Algunas consideraciones para orientar la enseñanza

La enseñanza debe ser planificada, es decir, pensada. Resulta fundamental que una actividad tan importante como la transmisión de la cultura a las nuevas generaciones sea planeada y planificada. Por más experiencia y creatividad que

posea el docente, la enseñanza no puede quedar librada a la improvisación. El docente reflexiona sobre su actividad profesional y prever sus acciones teniendo presentes las particularidades de sus estudiantes y del contexto, en función de las características del objeto de conocimiento que pretende transmitir (Davini, 2008).

Cualquiera sea la modalidad de planificación de la enseñanza (proyecto, problemas, casos) y el tipo de escritura que se adopte, Davini (2008) sostiene que el hecho de planificar la enseñanza implica necesariamente:

- Clarificar los propósitos educativos y definir los objetivos de aprendizaje específicos.
- Organizar los contenidos.
- Diseñar la estrategia particular de enseñanza, apoyándose o integrando aportes de distintos métodos, pertinentes para los objetivos planteados.

Orientaciones para el diseño proyectos interdisciplinarios que incluyan el pensamiento computacional

Para la realización del diseño de proyectos interdisciplinarios consideramos importante recuperar algunas ideas de Resnick y Silverman (2015), Papert (2001) y Ruck (2012) que sirvan de guía:

- Diseñar proyectos.
- Planificar actividades de piso bajo, techo alto y paredes anchas.
- Trabajar por desafíos.
- Habilitar diferentes estilos de trabajo y formas de resolución.
- Alentar el trabajo colaborativo y las alianzas para el aprendizaje.
- Tratamiento del error y las emociones.
- Evaluación formativa.

La Matriz de Seguimiento

Una herramienta que nos permite orientar los proyectos en la escuela es una matriz en donde se describen con detalle los elementos que componen al proyecto y las variables que indican cuáles de esos elementos están presentes y cuáles se podrían agregar para mejorar la propuesta.

En esta actividad deberán utilizar la dimensión 2 "Las Ciencias de la computación y el desarrollo curricular" de la Matriz TIC elaborada por la especialista en tecnología educativa María Teresa Lugo para analizar el caso presentado en la

Actividad 1. Seguramente solo podrán identificar algunos de los ítems de la Matriz en los casos dados debido a que no contamos con la información institucional que nos permita comprender el caso en todas sus dimensiones. Cuando no tengan dato para dar cuenta de un ítem de la Matriz, simplemente coloquen “sin dato”.

La matriz no es un instrumento de evaluación, sino una herramienta que nos permite entender cuáles son los elementos centrales que necesita un proyecto que integra la computación en la escuela y cómo podemos aportar para que los docentes tengan los recursos y apoyos necesarios para un proyecto que promueva aprendizajes significativos.

La Matriz permitirá:

- a) conocer cuáles son las variables indispensables a considerar a la hora de llevar adelante una política de inclusión de Ciencias de la Computación en el aula.
- b) saber cuáles son las expectativas deseables en relación al tema según el grado de avance en relación a la posible innovación en cada escuela
- c) situar a una escuela en relación a estas variables y sus indicadores
- d) diseñar estrategias de acompañamiento según el momento de cada escuela en relación a la innovación.

Asimismo, realicen un análisis cualitativo del caso a la luz de los aportes ofrecidos en esta clase (de dos carillas).

Actividad 3: Focalicen en la dimensión 2 de la matriz y planifique acciones pedagógicas para lograr el nivel avanzado que propone.

Actividad 4: Sincrónica. Compartimos las planificaciones

EVALUACIÓN

Para aprobar este módulo es requisito tener aprobado el 80% de las tareas prácticas y el trabajo final.

BIBLIOGRAFÍA

Benítez Larghi, S, Aguerre, C; Calamari, M; Fontecoba, A; Moguillansky, M (2013). De brechas, pobrezas y apropiaciones. Juventud, Sectores Populares y TIC en la Argentina. Versión. Disponible en http://www.academia.edu/1115074/De_brechas_pobrezas_y_apropiaciones._Juventud_Sectores_Populares_y_TIC_en_la_Argentina

Bonello, M.B, Schapachnik, F. (2020). "Diez preguntas frecuentes (y urgentes) sobre pensamiento computacional". VEC. Año 11, Vol. 11, Núm. 20.

<https://revistas.unc.edu.ar/index.php/vesc/article/view/27453>

Caraballo, S., & Cicala, R. (2005). Hacia una Didáctica de la Informática... Ponencia presentada en las Primeras Jornadas Nacionales en Didácticas Específicas. Organizadas por UNSAM. https://edutice.archives-ouvertes.fr/file/index/docid/285062/filename/a0601c_esp.htm

CFE (2018). Núcleos de Aprendizajes Prioritarios (NAP) de Educación Digital, Programación y Robótica. Ministerio de Educación, Cultura, Ciencia y Tecnología. Lea principalmente los del nivel educativo en donde se desempeña.

<https://www.educ.ar/sitios/educar/resources/150123/nap-de-educacion-digital-programacion-y-robotica/download>

Davini, M. C. (2008). Métodos de enseñanza. Didáctica general para maestros y profesores. Buenos Aires: Santillana.

Entrevista a Susana Muraro: "La Informática perdida"

<http://lainformaticaprohibida.blogspot.com/2013/04/hoy-la-entrevistada-es-susana-muraro.html>

Levis, D., & Cabello, R. (Eds.). (2007). Medios informáticos en la educación a principios del siglo XXI. Prometeo Libros Editorial.

Manuales de Ciencias de la computación para el aula de la Fundación Sadosky

Palma Suárez, C.; Sarmiento Porras, R. (2015). Estado del arte sobre experiencias de enseñanza de programación a niños y jóvenes para el mejoramiento de las competencias matemáticas en primaria. En Revista mexicana de investigación educativa, vol. 20, no. 65. Disponible en: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-66662015000200013

Papert, S. (1995). La máquina de los niños: replantearse la educación en la era de los ordenadores. Barcelona: Paidós. Disponible en: <https://mahara.org/artefact/file/download.php?file=75625&view=17519>

Sage, S., & Torp, L. (1999). Aprendizaje basado en problemas, El: Desde El Jardín de Infantes hasta El Final de La Escuela Secundaria.

Simari, G. R. (2013, July). Los fundamentos computacionales como parte de las ciencias básicas en las terminales de la disciplina Informática. In VIII Congreso de Tecnología en Educación y Educación en Tecnología.

Intendente
Dr. Martín Llaryora

Viceintendente
Dr. Daniel Passerini

Secretaría de Educación, Cultura e Innovación
Dr. Horacio Ferreyra

Dirección General de Educación
Lic. Pablo Rodríguez Colantonio

Dirección de Gestión Educativa
Lic. Luis Franchi

Subdirección de Innovación Educativa y Desarrollo Profesional Docente
a/c Subdirección de Aprendizaje y Curriculum
Dra. Alicia Olmos

Subdirección de Coordinación Operativa
a/c Subdirección de Fortalecimiento Socioeducativo
Lic. Alicia La Terza



Secretaría de
Educación



Municipalidad
de Córdoba