

Innovación educativa y gestión del conocimiento

Oscar Yecid Aparicio Gómez y Olga Lucía Ostos Ortiz
Editores académicos



Enseñanza del pensamiento computacional: metodologías y tecnologías educativas

XABIER BASOGAIN OLABE
MIGUEL ÁNGEL OLABE BASOGAIN
JUAN CARLOS OLABE BASOGAIN

El pensamiento computacional está abriéndose camino en los centros escolares y en la sociedad en general, gracias al uso de metodologías y tecnologías educativas innovadoras. Los recursos tecnológicos facilitan el ingreso del pensamiento computacional a la educación formal. Además, un uso eficaz de las metodologías soportadas por la propia tecnología educativa permite mejorar la enseñanza del pensamiento computacional en ámbitos de educación no formal. Este capítulo no trata sobre el concepto y desarrollo del pensamiento computacional (descrito y tratado en muchos artículos académicos), sino que presenta el recorrido que el grupo de investigación docente al que pertenecen sus autores ha realizado en los últimos doce años, para llevar al aula y a la sociedad la educación y formación del pensamiento computacional.

Este recorrido recoge muchas experiencias en diferentes formatos: proyectos, cursos, talleres, conferencias, artículos, exposiciones, colaboraciones internacionales, apoyos institucionales y lecciones aprendidas. El propósito de esta exposición de experiencias es ofrecer a futuros emprendedores de la educación del pensamiento computacional (y otras

materias afines como STEM/STEAM: Science, Technology, Engineering, Arts and Mathematics) una referencia que les ayude a tomar decisiones para lograr sus objetivos educativos del pensamiento computacional en el aula.

Introducción

La historia del pensamiento computacional (PC), como es el caso en muchos desarrollos fundamentales en la ciencia, refleja la convergencia de múltiples ideas provenientes de diferentes áreas de estudio como ciencias cognitivas, lingüística, psicología y ciencias de la computación. Estas áreas, después de desarrollarse de manera aislada, encuentran un efecto sinérgico cuando se aplican al área de la educación, en particular, a los procesos que involucran lenguajes generativos para la creación de métodos novedosos y sistemas complejos.

Algunos de los pensadores pioneros en este campo son Papert, Wing y Wolfram. Una de las primeras referencias al PC se encuentra en Disessa (2000), donde describe el valor de aplicar primitivas soluciones cognitivas humanas a problemas orientados a objetos, al observar las relaciones entre los componentes de un sistema complejo. Se pueden encontrar otras referencias similares en Attwell (2010) y Aprendiendo Scratch (2015), donde se exponen las ideas fundamentales de dividir una tarea compleja en un conjunto de tareas más simples.

Podemos encontrar descripciones específicas con respecto a los elementos centrales del PC en los trabajos de dos asociaciones, The International Society for Technology in Education (ISTE) y The Computer Science Teachers Association (CSTA), que en colaboración con la industria y la educación K-12, han sido desarrolladas para atender las necesidades de los educadores que integrarán la disciplina del PC en las aulas. Las principales características del PC son las siguientes (Benfield, Roberts y Francis, 2006): 1) permite formular problemas que pueden ser resueltos por una computadora y otras herramientas; 2) organiza y analiza datos de manera lógica; 3) representa datos a través de abstracciones como modelos y simulaciones; 4) sugiere soluciones automatizadas a través del pensamiento algorítmico (una serie

de pasos ordenados); 5) identifica, analiza e implementa posibles soluciones con el objetivo de lograr la combinación más eficiente y efectiva de pasos y recursos; 6) generaliza y transfiere estos procesos de resolución de problemas a una amplia variedad de dificultades diferentes.

El campo del PC todavía está en una etapa inicial, en el proceso de definir sus objetivos generales y sus principios fundamentales. Además, tiene dos grandes temas que deben ser objeto de investigación en un futuro próximo: capacidades cognitivas y experiencias creativas. En primer lugar, es preciso descubrir el mundo inexplorado de los procesos cognitivos que la mente humana puede soportar, siempre que se proporcionen las herramientas necesarias. La ciencia cognitiva ha alterado dramáticamente el modelo para el cual se diseñó el modelo educativo del pasado. Ahora sabemos mejor que antes cuáles son los límites de la mente (Aprendiendo Scratch, 2015; Berkeley University, 2017; Breslow *et al.*, 2013; Bruff *et al.*, 2013), la tarea que queda por delante es descubrir los recursos computacionales, primitivos, de la mente, para cosechar su poder. En segundo lugar, es importante desarrollar un conjunto de experiencias que les permita a los estudiantes adquirir estas herramientas, participar en estos procesos cognitivos y producir resultados innovadores. El sistema educativo K-12 y la educación superior se basan en un currículo con uso generalizado de lenguajes descriptivos y problemas de tipo A (College Board, 2016). El plan de estudios que responda a las necesidades futuras se diseñará para el uso de lenguajes generativos (lenguajes orientados a objetos) y problemas de tipo B. Esta tarea requerirá el esfuerzo y la colaboración de todos los campos que participan en el PC.

Un ejemplo de sistema educativo donde se llevan a cabo algunos de estos pasos prácticos para integrar el PC en el aula es el de Inglaterra en el Reino Unido. En este país, el currículo nacional en PC aborda los objetivos de educar a sus estudiantes para que puedan: 1) comprender y aplicar los principios y conceptos fundamentales de la informática, incluidos la abstracción, la lógica, los algoritmos y la representación de datos; 2) analizar problemas en términos computacionales y tener experiencia práctica en escribir programas de computadora para resolver tales problemas; 3) evaluar y aplicar tecnología de la información, incluidas tecnologías nuevas o desconocidas, para resolver problemas

analíticamente; y 4) ser usuarios responsables, competentes, confiables y creativos de la tecnología de la información y la comunicación (Coursera, 2015). En el caso de los Estados Unidos, el camino predominante para que los estudiantes preuniversitarios participen en las actividades y el plan de estudios de PC son los cursos de ciencias de la computación de Advanced Placement. Estos cursos se ofrecen en escuelas secundarias y, después de graduarse, los estudiantes reciben créditos universitarios para cursos introductorios de informática. Algunos desarrollos recientes (Department for Education England, 2013; edX, 2015) han extendido la inscripción de los cursos de PC a estudiantes universitarios de otros campos.

El capítulo presenta dos secciones, la primera expone proyectos educativos sobre PC y la segunda aborda metodologías y tecnologías educativas aplicadas a la enseñanza del PC. Los autores, profesores de universidades del País Vasco y de Estados Unidos, son miembros de un grupo de investigación del sistema universitario vasco que realiza su trabajo en las áreas de educación, tecnología educativa y computación.

Proyectos educativos sobre PC

A continuación presentamos un conjunto de proyectos que muestran la evolución del modelo de enseñanza del PC, en un contexto en el que la tecnología educativa se ha transformado y ofrece nuevos servicios y prestaciones para desarrollar con mayor éxito el proceso de enseñanza del PC.

LearnScratch y AprendiendoScratch

El portal web LearnScratch.org fue creado en el verano de 2007 como un recurso educativo para ayudar a los maestros de habla inglesa a incorporar Scratch (<http://scratch.mit.edu>) en el aula. Se basaba en un completo conjunto de video tutoriales que presentaban un camino para explorar las posibilidades de Scratch, y una descripción detallada de proyectos sobre dibujo, música, matemáticas, narración, animación, etc., que servían como una caja de herramientas inicial de proyectos

para expandir, personalizar y explorar experiencias utilizando este entorno de desarrollo.

A petición de los maestros que usaron el portal web, en el verano de 2008, los materiales se reestructuraron en tres cursos separados en función de su público, objetivo y la complejidad y el tiempo necesarios para completar las lecciones. Cada curso se acompañaba de un conjunto de lecciones que describían los objetivos, el contenido y los proyectos relacionados para cada una.

El proyecto AprendiendoScratch nació como la realización de uno de los objetivos establecidos en LearnScratch.org que se presentaron a la comunidad Scratch en la conferencia Scratch @ MIT2010 (Glance, Forsey y Riley, 2013).

El reto para nuestro grupo fue la creación de una serie de recursos académicos diseñados para incorporar Scratch en el currículo de las escuelas de habla hispana, para la educación primaria y secundaria.

El primer paso para cumplir este reto fue construir el portal AprendiendoScratch.org, lanzado en julio de 2011 y difundido en la comunidad virtual de Educadores de ScratchEd (Seliskar, 2014). El portal contenía una colección de ocho cursos de video correspondientes a las ocho familias de componentes básicos de Scratch versión 1.4. Cada curso se organizaba en un conjunto de unidades o lecciones (tres a siete, dependiendo de la complejidad de la familia de bloques), y cada lección a su vez constaba de seis proyectos de Scratch.

Cada proyecto tenía asociado un video tutorial que describía en detalle sus objetivos y funcionamiento.

El equipo AprendiendoScratch creó además una colección inicial de ocho libros en castellano y el libro *HelloScratch* en inglés, que incluían todos los proyectos desarrollados en los video tutoriales de ambos portales. El objetivo principal en la publicación de la colección fue proporcionar a los maestros y estudiantes un conjunto de recursos en formato impreso para facilitar la implementación de los proyectos y complementar los recursos en línea. Cada proyecto incluía los textos de su código, una breve descripción de sus objetivos y operación, e imágenes del trabajo final.

El proyecto educativo AprendiendoScratch incluía además del portal web y los libros, otras actividades: 1) talleres de capacitación, algunos en formato de aprendizaje virtual; 2) visitas a centros educativos e instituciones gubernamentales de educación; y 3) difusión en foros académicos y científicos, que describiremos más adelante.

Nuestro equipo también realizó para ScratchJr (<https://www.scratchjr.org/>) un conjunto de criterios de diseño y de guías para el desarrollo de materiales curriculares y métodos de evaluación (Olabe, Basogain y Olabe, 2017). ScratchJr está inspirado en Scratch y fue desarrollado para incorporar la programación en las aulas de los grados K-2 (de edades entre 5 y 7 años). La consecución de este objetivo requiere la creación de un currículo apropiado para esa edad y un conjunto de herramientas de evaluación correspondientes.

Aproximadamente 2000 escuelas de más de 150 países siguieron los video tutoriales de los portales LearnScratch y AprendiendoScratch. En aquellos años, la enseñanza de PC se realizó de una manera unidireccional, basada en video tutoriales y libros de texto. Los dos portales educativos LearnScratch.org y AprendidoScratch.org están fuera de línea desde el año 2015.

Proyecto Pensamiento Computacional en la Escuela, Miríadax

El curso titulado “Pensamiento Computacional en la Escuela” fue el primer curso MOOC (Massive Open Online Course) ofrecido por la Universidad del País Vasco/Euskal Herriko Unibertsitatea en la plataforma Miríadax (<https://miriadax.net>). Esta es una de las mayores plataformas de cursos MOOC del mundo, con más de cuatro millones de usuarios en la actualidad.

El objetivo del curso fue definir y utilizar el PC en la escuela y en el día a día. El PC conlleva analizar en términos computacionales el mundo que nos rodea, y aplicar herramientas y técnicas de computación para entender y resolver las diferentes situaciones que se produzcan en la vida cotidiana. En este sentido, el curso fue diseñado para lograr el objetivo propuesto utilizando los recursos MOOC que ofrecía la plataforma Miríadax. El curso estaba orientado a profesores y

personas interesadas en aprender a resolver problemas a través del uso del lenguaje de programación Scratch.

Las dos primeras ediciones del curso se realizaron en los periodos académicos 2014-2015 y 2015-2016 con estudiantes de 35 países. La experiencia de este curso mostró la posibilidad de impartir un programa masivo, virtual y abierto sobre PC, así como la necesaria combinación de tecnologías y metodologías didácticas para realizar un curso de esta naturaleza con éxito (Singh, 2015).

Los MOOCs son una alternativa a la demanda actual de aprendizaje permanente. Cada vez más la formación convencional basada en cursos presenciales, talleres y seminarios está dando paso a otras formas de aprender, y en particular al aprendizaje a través de la red. Ya es una realidad que la formación del profesorado y la formación en general han comenzado a convivir con los cursos denominados MOOC.

El planteamiento de este curso tuvo en cuenta: 1) las características, masivo y en línea que presenta la audiencia de un curso MOOC; y 2) la existencia de otros cursos que tratan la temática de la programación y la computación para jóvenes estudiantes y docentes, como “Code yourself! An introduction to programming” en Coursera, “CS002x programming in Scratch” en EdX y “Creative computing online workshop” en Harvard Graduate School of Education (Kahneman, 2003; 2011; Lu y Law, 2012).

La metodología del curso siguió las directrices de un curso MOOC. Esto implicaba que el alumno debía realizar las siguientes acciones: 1) seguía una serie de video-lecciones cortas; 2) realizaba cuestionarios interactivos; 3) era evaluado a través de evaluaciones de los compañeros p2p (*peer to peer*); y 4) estaba en contacto virtual con compañeros y profesores.

El curso puso especial énfasis en la experiencia del aprendizaje en línea del alumno. Para ello destacamos tres aspectos relevantes del diseño del curso: 1) contenidos multimedia y actividad Scratch, 2) autotest y 3) tareas (p2p).

Las tareas p2p desempeñan un papel relevante en los cursos MOOC. Se trata de un recurso pedagógico que está evolucionando con nuevas prestaciones a medida que se comprueba su validez y eficacia. El curso utilizaba de forma sistemática este recurso, y destacamos dos

características de las tareas p2p: responsabilidad en el proceso de evaluación y otras facetas del evaluador.

Otro aspecto relevante a destacar fue la creación y mantenimiento de la comunidad de participantes del curso. En general, un curso MOOC dispone del foro como un mecanismo de comunicación entre los alumnos. La idea inicial de que el foro es simplemente un medio de comunicación se transforma a medida que el curso está en marcha. El foro pasa a ser el espacio de comunicación y el lugar donde la comunidad se hace, allí los estudiantes comparten dudas y dificultades, realizan consultas, dan consejos, expresan comentarios y reflexiones (Johannesen, 2013). A lo largo del curso, semana tras semana, aparecen problemas y situaciones individuales que una vez expresados en el foro pasan a ser abordados por la comunidad. Este proceso tiene una realimentación positiva en todos los participantes del curso, independientemente del grado de participación que tengan en el foro.

En el panel digital titulado “1ª Edición del curso Pensamiento Computacional en la Escuela” se recogen los principales componentes del curso: video tutoriales, proyectos Scratch, autotest, test, foros, organización, certificado de participación y testimonios de los participantes.

Más de 4000 alumnos de 35 países se inscribieron en el curso, esto sugirió lo siguientes interrogantes: ¿cómo medir la tasa de éxito del curso?, ¿por el número de participantes que iniciaron y finalizaron?, ¿por los testimonios que dan al finalizar el curso? La modalidad MOOC facilita la enseñanza de PC de una manera bidireccional, activa, en formato curso (objetivos, calendario, tareas) y con registro del progreso de los participantes.

Proyecto PC-01: introducción al PC

En este proyecto presentamos un curso sobre PC para educación primaria y secundaria donde se enseñan los “conceptos” y “procesos” básicos de computación para jóvenes estudiantes de 10 a 12 años. El curso está diseñado para ser impartido en formato híbrido o mixto de educación. El profesor está presente con los estudiantes en el aula y utiliza una plataforma de aprendizaje en línea (Mak, Williams, y

Mackness, 2010). La informática educativa nos ofrece cada día nuevos productos y herramientas pedagógicas que facilitan la tarea de enseñar materiales que hasta ahora no se podían incluir en el currículo escolar (Misión Scratch, 2010).

Las plataformas de aprendizaje o los entornos virtuales de aprendizaje (VLE) ofrecen beneficios a maestros y estudiantes. Los maestros fomentan la colaboración y la comunicación, y pueden personalizar y diferenciar el progreso en el aula. Los estudiantes, por su parte, aprenden a trabajar con sus compañeros en proyectos y a desarrollar habilidades de colaboración y resolución de problemas.

Los VLE también presentan algunos inconvenientes que incluyen, entre otros, la dificultad intrínseca del uso de la tecnología, la falta de apoyo técnico y pedagógico para los docentes y el requisito de una decisión formal y de una inversión económica por parte de la institución educativa. También hay algunos conceptos erróneos con respecto a los VLE, como considerarlos una alternativa a las clases tradicionales, o asumirlos exclusivamente como una modalidad de educación en línea (Olabe, Basogain, y Olabe, 2017).

Los VLE se han desarrollado en todos los niveles de educación con diferentes grados de implementación. En la educación superior, la implementación es extensa, mientras que en la secundaria se ha limitado a proporcionar la infraestructura tecnológica, y en la primaria, con muy pocas excepciones, es casi inexistente (Olabe, Basogain, y Olabe, 2015; 2016a; 2016b).

Este panorama está comenzando a cambiar como resultado de varios desarrollos: 1) las escuelas públicas y privadas están descubriendo el valor educativo de este nuevo tipo de aprendizaje que incluye multimedia, evaluación, monitorización y colaboración; 2) desde un punto de vista técnico, hay una evolución hacia VLE que los hace más fáciles de usar y los adapta a las necesidades de los estudiantes, al hacer más sencilla la instalación y mantener la información en la memoria de los computadores o en la nube; y 3) el fenómeno MOOC (Olabe, Basogain y Olabe, 2010; 2011) y sus variaciones (COOC, NOOC, SPOC) (Olabe, Basogain y Olabe, 2014; 2016; 2017), que han revolucionado la forma en que usamos la tecnología en educación a distancia y mixta.

El curso “PC-01: introducción al pensamiento computacional” está diseñado con las siguientes características: 1) implementación inmediata en la escuela; 2) acceso simple a los contenidos y herramientas por parte del profesor y los alumnos; 3) introducción básica a conceptos y procesos en PC; y 4) uso eficiente y sostenible de la tecnología educativa.

El curso se desarrolló en la plataforma de aprendizaje Egelapi (<https://egelapi.ehu.eus/>), que es un VLE basado en Moodle. El curso utilizó el *software* Scratch 2.0 como entorno de lenguaje de programación (editor en línea y editor fuera de línea de Scratch 2). Durante el curso se incluyó el estudio de los siguientes elementos del PC: 1) pensamiento y expresión computacional (cómo leer y escribir en un lenguaje formal para resolver problemas); 2) abstracción (cómo comunicar ideas complejas de manera simple y, lógicamente, descomponer los problemas); 3) integración de contenido multimedia (texto, imágenes, sonido, datos, gráficos); 4) desarrollo de objetos y bloques funcionales (objetos, programas); 5) programas interactivos (eventos y gestión de eventos); y 6) conceptos fundamentales de programación (decisiones, bucles, variables, funciones, ejecución secuencial y paralela).

El curso se organizó en diez sesiones, cada una de dos horas. Las sesiones se nombraron de acuerdo con las familias de bloques de Scratch que se estudiaron: Movimiento, Apariencia, Sonido, Lápiz, Evento, Control, Sensor, Operadores, Datos y Más bloques. Además, se creó una sesión inicial para familiarizar al estudiante con la plataforma de aprendizaje Moodle y el lenguaje de programación Scratch.

Los recursos didácticos y la metodología que se utilizaron en todas las sesiones del curso tenían una estructura común con los siguientes elementos básicos descritos en la tabla 1.

Tabla 1. Recursos didácticos y metodológicos del curso integrados en la plataforma

Símbolo	Herramienta	Descripción
	Video	Conjunto de cuatro a seis tutoriales en video (de 3 a 5 minutos cada uno) en el que se presentan los conceptos de la sesión.
	Práctica	Plantilla de proyecto Scratch para permitir al alumno explorar el proyecto presentado durante el video tutorial. Si es necesario, es posible volver a visitar el video tutorial para lograr una comprensión completa de la sesión.
	Auto Test	Autoevaluación (5 minutos) por parte del alumno para determinar el grado de conocimiento adquirido (se puede repetir tantas veces como sea necesario).
	TEC (tarea evaluada por un compañero)	Tarea realizada por el alumno para resolver un problema mediante la creación de un proyecto Scratch. Los proyectos son evaluados por compañeros estudiantes utilizando una rúbrica común. En inglés, el término p2p se utiliza para identificar este tipo de evaluación por pares.
	Test	Prueba evaluativa (5 minutos) que mide el grado de conocimiento adquirido por el estudiante (2 intentos).
	Explorar y descubrir	Proyectos Scratch para que los estudiantes amplíen sus conocimientos, descubran y exploren nuevas formas de usar los bloques Scratch.

Fuente: elaboración propia.

La metodología del curso se basó en un paradigma centrado en el alumno, por tanto, las tareas del estudiante incluían: 1) seguir una serie de lecciones cortas en video, 2) evaluar y ser evaluados a través de test y p2p; y 3) participar en foros con compañeros y profesores (Papert, 1996).

Este curso enfatizó la experiencia de aprendizaje de los estudiantes. En este sentido, vale la pena destacar tres aspectos importantes del diseño del curso: 1) contenido multimedia y actividad Scratch; 2) auto-test; y 3) tareas p2p. Los “conceptos” y los “procesos” centrales del pensamiento computacional (Pinker, 1995) que se desarrollaron en el curso se evaluaron en las actividades de Test y TEC.

El profesor disponía de un conjunto de recursos y servicios que facilitaban la entrega y el monitoreo del curso introductorio en PC. Además, el profesor tenía acceso a una colección de video tutoriales, auto-test, tests y TECs que cubrían la agenda básica de las diez familias de los bloques de Scratch. Antes de comenzar el curso, el profesor recibía capacitación sobre la metodología y el contenido del curso.

El curso incluía 50 tutoriales en video con una duración de 3 a 6 minutos cada uno, un banco de 120 preguntas de opción múltiple utilizadas en auto-test y test, y un conjunto de 10 proyectos de Scratch que los estudiantes realizaban y evaluaban en forma de TEC.

Los TEC utilizaban el módulo de actividad “Taller” en Moodle, que permite la recopilación, la revisión por pares y la evaluación del trabajo de los alumnos. La programación de este módulo implicaba completar las fases de configuración, envío de proyectos, evaluación y calificación de las evaluaciones (Pinker, 1999).

Los profesores también disponían de una serie de herramientas de aprendizaje analítico integrado en el VLE (Moodle), que les permitía implementar evaluaciones del progreso de los estudiantes, y pruebas colectivas e individuales de “conceptos” y “procesos” en el PC (Project Microworlds, 2017).

Del mismo modo, el profesor tenía acceso a los registros de las calificaciones totales y parciales de los estudiantes. Las herramientas de gestión permitían agrupar las tareas en diferentes colecciones para obtener información relevante sobre la finalización y el progreso del curso.

El curso PC-01 se caracterizó por el hecho de que los estudiantes construyeron su propio conocimiento siguiendo la teoría pedagógica del construccionismo. Esta teoría, propuesta por Seymour Papert, propone que los estudiantes pueden desarrollar sus conocimientos a través de la construcción de un “artefacto” que los motive. Para la construcción de un “artefacto” el curso dedicaba dos secciones, llamadas “Práctica” y “TEC”, donde se utilizaba el entorno de programación Scratch.

En la sección práctica, los alumnos experimentaban con Scratch los contenidos conceptuales desarrollados por el profesor en los tutoriales en video; en la sección TEC, los estudiantes debían diseñar y construir un proyecto Scratch para resolver una situación o un problema propuesto utilizando los contenidos y conceptos aprendidos en la sesión.

Uno de los objetivos del curso era que los estudiantes aprendieran los principios de un lenguaje de programación de ordenadores. En este curso, los estudiantes usaron Scratch como una herramienta que les permitía comunicarse y crear ideas para resolver problemas (Proyecto PC-01, 2018).

La implementación de un proyecto Scratch que resuelve un problema requiere el desarrollo de funciones cognitivas superiores de la mente. Estas funciones incluyen categorización, toma de decisiones, abstracción, conocimiento, resolución de problemas, planificación y ejecución.

La segunda parte de las tareas de la sección TEC, que consistía en la evaluación de los proyectos Scratch de tres a cinco compañeros de estudios, promovía también el desarrollo de funciones cognitivas superiores. Esta fase se realizaba de forma anónima y siguiendo una rúbrica específica. Los estudiantes debían desempeñar el papel de evaluadores y, a su vez, profundizar sus conocimientos para evaluar los proyectos Scratch creados por otros estudiantes.

El VLE del curso permitía al estudiante no solo desempeñar el papel de evaluador, sino también colaborar con sus compañeros al dar y recibir comentarios, indicando errores, soluciones y posibles mejoras.

Además de estar expuestos a otros proyectos Scratch, estas actividades le daban al estudiante la oportunidad de aprender y ser inspirados por nuevas soluciones para problemas futuros (Paddick, 2014).

La experiencia de los estudiantes en este curso híbrido PC-01 también les permitió familiarizarse con las herramientas y metodologías del VLE, al experimentar diferentes aspectos de esta forma de aprendizaje, tales como: contenido en formato multimedia, mecanismos de colaboración, autoevaluación y evaluación, progreso en sus conocimientos y calificaciones. Cabe mencionar que el curso ofrecía una insignia, un premio o medalla, para reconocer los logros y brindar a los estudiantes una recompensa que reconoce sus logros en PC (Recursos en español, 2017).

El curso PC-01 es un ejemplo real de sinergia entre la informática y la tecnología educativa. En su desarrollo, el curso utilizó el potencial educativo de los VLE para formar estudiantes de primaria y secundaria en la nueva asignatura curricular del pensamiento computacional. Asimismo, explotó las herramientas analíticas de aprendizaje y le permitió al profesor realizar con éxito las tareas de enseñanza, monitoreo y calificación de los estudiantes. Por su parte, los estudiantes tuvieron una experiencia de aprendizaje en un entorno constructivo de colaboración y se mostraron capaces de monitorear y seguir su propio progreso.

Proyectos de colaboración internacional

Nuestro grupo de investigación ha establecido relaciones internacionales con los ministerios de educación de Perú y de República Dominicana, así como con entidades adscritas al proyecto One Laptop per Child (OLPC) de Uruguay, Argentina, México, Paraguay y Perú. Asimismo, el grupo ha participado en un proyecto para introducir el PC en las escuelas de Chile, con la colaboración de agentes innovadores, entre los que se encuentran la Fundación Telefónica Chile, UCORP, Universidad de Concepción, INACAP y el Ministerio de Educación de Chile (Mineduc).

La colaboración con estas instituciones se ha desarrollado en diferentes modalidades que incluyen visitas a centros escolares, conferencias divulgativas, entrega de material docente en formato electrónico, talleres formativos a educadores y asistencia al desarrollo del curso PC-01.

El proyecto AprendiendoScratch incluyó el diseño y la realización de talleres de capacitación presenciales, así como espacios de formación virtuales. La primera serie de estos talleres se desarrolló en un proyecto de colaboración con OLPC (Papert, 1991; Sancho y Padilla, 2016) en América Latina. Estos talleres se ofrecieron a maestros, instructores y capacitadores de Perú, Paraguay, México, Uruguay y Argentina con entrega de material docente (Sykora, 2014).

El equipo del proyecto AprendiendoScratch ha visitado muchos centros educativos en América Latina. Durante estas visitas, se entrevistó a estudiantes, profesores y autoridades institucionales.

El curso PC-01 se ha impartido en más de veinte centros educativos de República Dominicana y de Colombia en colaboración con los respectivos ministerios de educación de dichos países y la Corporación Red Nacional Académica de Tecnología Avanzada (Renata) de Colombia.

Los resultados del curso se evaluaron utilizando el portafolio de cada estudiante y el número de exámenes superados. Estos parámetros indicaron un alto grado de éxito. Otros resultados del curso se obtuvieron de entrevistas personales con estudiantes y profesores, y mostraron un alto grado de satisfacción por parte de los participantes del curso (entre las expresiones más utilizadas para describir el curso se encontraron: fácil, divertido, entretenido, aprender más, crear).

El curso PC-01 está disponible para cualquier iniciativa educativa que tenga como objetivo introducir el PC en las escuelas primarias y secundarias en todo el mundo. Las instituciones educativas interesadas en implementar el curso PC-01 pueden ponerse en contacto con los autores del presente capítulo para establecer una relación de cooperación internacional que promueva la innovación educativa.

Tecnologías y metodologías

Esta sección presenta un conjunto de experiencias en las que se incorporan nuevos desarrollos tecnológicos como el entorno de programación Snap (<https://snap.berkeley.edu/>) y diferentes sistemas de web-conferencia que mejoran la enseñanza y divulgación del PC.

Ecosistemas con Snap

Los estudiantes que tienen poca o ninguna experiencia en las áreas científico-técnicas del PC y de STEAM tienen, en general, la percepción de que cualquier problema es complejo, más allá de sus capacidades cognitivas. Además, estos estudiantes a menudo carecen de motivación para abordar este tipo de problemas que consideran complejos. Esta es una barrera psicológica que históricamente ha limitado el éxito académico de los proyectos de PC en el aula. Para superar esta barrera psicológica desarrollamos una estrategia basada en la creación de entornos de programación que etiquetamos como ecosistemas computacionales.

Estos ecosistemas son diseñados de tal manera que, a pesar de tener una alta complejidad, permiten a los estudiantes asimilarlos con familiaridad mediante el uso de un andamio cognitivo estructurado en torno a metáforas computacionales que los estudiantes han adquirido a través de experiencias en diversas áreas de su vida cotidiana.

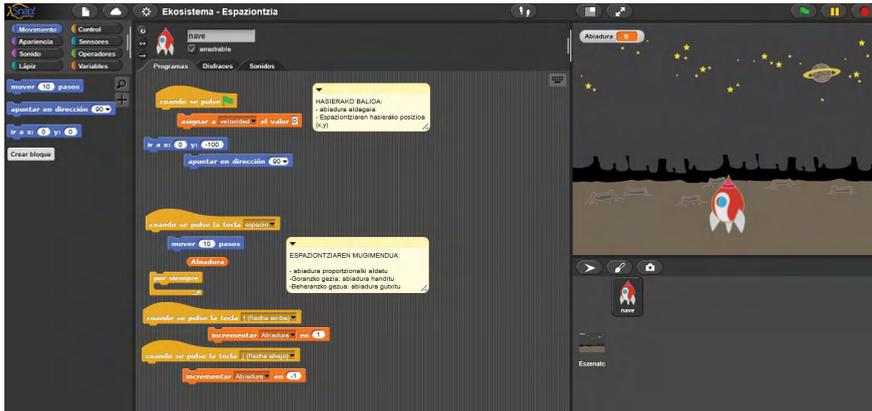
Cada ecosistema se implementa con una versión separada y personalizada de Snap. El ecosistema contiene el conjunto de bloques de programación necesario para la creación y el desarrollo de todas las actividades de los estudiantes. Esta versión personalizada de Snap tiene el beneficio directo de identificar para los estudiantes aquellos elementos computacionales que son relevantes para el proyecto. El número de bloques de programación iniciales en Snap, que son más de 120, se reduce a una docena de bloques aproximadamente, al usar esta estrategia. Los estudios informales muestran que los estudiantes responden a este tipo de entorno con mayor optimismo.

Por ejemplo, un ecosistema que combina las reglas de un juego con las leyes de la física implementa un cohete espacial que el estudiante debe pilotar. Las leyes físicas que controlan el cohete se incluyen directamente en códigos (programas) simples que el estudiante modifica y diseña.

Debido a que el estudiante debe pilotear el cohete y hacer que despegue y aterrice de manera correcta, aprender las relaciones que gobiernan el comportamiento de este sistema proporcionará motivación, ya que su interés es pilotear el cohete sin destruirlo. La figura 1 muestra

el ecosistema computacional diseñado para el vuelo de un cohete espacial realizado en el entorno personalizado de Snap (Wolfram, 2016).

Figura 1. Ecosistema computacional para el vuelo de un cohete espacial controlado por las leyes de Newton



El equipo de Scratch del MIT está diseñando “micromundos” que son versiones personalizadas del editor de Scratch que contienen un pequeño conjunto de bloques para hacer proyectos basados en un tema. Los micromundos ofrecen un punto de entrada más creativo a la programación al brindar una experiencia abierta, que permite a los niños explorar, experimentar y crear, a la vez que proporcionan un punto de entrada más simplificado y escalonado en la programación (Basogain *et al.*, 2019).

Los ecosistemas computacionales o micromundos favorecen la participación de los estudiantes en los procesos de programación porque el entorno presenta un andamiaje cognitivo que facilita la consecución del proyecto Snap o Scratch.

Webinars

La tecnología nos ofrece servicios para realizar videoconferencias y reuniones virtuales que facilitan muchas de las actividades que realizamos de forma presencial. Ilustramos con tres ejemplos concretos el uso de la tecnología de herramientas síncronas para realizar tareas de

formación y tutoría en los proyectos: AprendiendoScratch, el curso MOOC en Miríadax y la participación en conferencias internacionales.

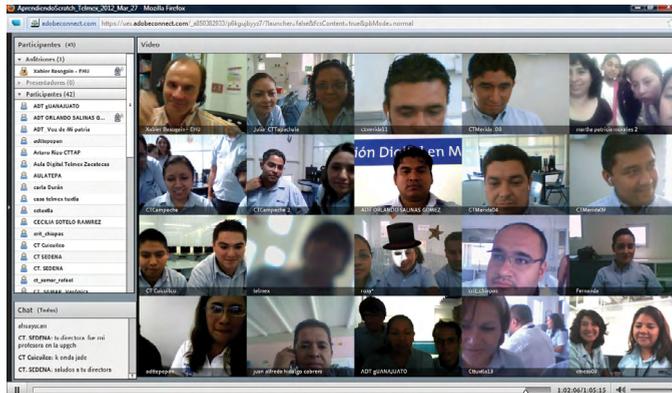
El primer ejemplo es del proyecto AprendiendoScratch, donde se impartió un taller virtual mensual a Telmex en México. El último martes de cada mes se ofreció el taller a través de un webinar a “Casas” y “Aulas Digitales” (<http://www.fundaciontelmextelcel.org/aulas-digitales-telmex>) de Telmex en más de veinte centros en todo México. Los temas tratados en cada webinar se desarrollaron durante el mes siguiente en las Aulas Digitales utilizando los recursos del portal AprendiendoScratch.

Las sesiones del seminario se realizaron en la plataforma de conferencia web Adobe® Connect™. Durante las diez sesiones de este seminario web participaron 4500 niños y 750 maestros en los proyectos Scratch propuestos por AprendiendoScratch. Además, este conjunto de grabaciones se ha integrado en un curso Moodle de la plataforma Telmex Education con todos los recursos utilizados durante los seminarios web.

La figura 2 muestra dos momentos de los webinars AprendiendoScratch (el ejemplo de grabación de webinar de la tercera sesión se encuentra disponible en: https://uex.adobeconnect.com/_a850382933/p6kgujbyyz7/).

Figura 2. Grabaciones de dos sesiones webinar AprendiendoScratch





El segundo ejemplo utiliza la herramienta Hangout On Air de Google (actualmente se llama YouTube Live) que ha sido utilizada para tener una sesión con participantes del curso de Pensamiento Computacional en las Escuelas de la plataforma Miríadax (grabación disponible en <https://youtu.be/H-T-vQsr01c>).

La grabación permite a los estudiantes que no pudieron asistir a la tutoría acceder a ella para estar informados del desarrollo de la sesión presencial en línea. La figura 3 muestra dos instantes de la tutoría realizada con la herramienta Hangout de Google.

Figura 3. Tutoría en línea del curso en Miríadax a través de Hangout On Air



Mesa Redonda + Q&A (curso Pensamiento Computacional en la Escuela)

Mesa Redonda + Q&A (curso Pensamiento Computacional en la Escuela)

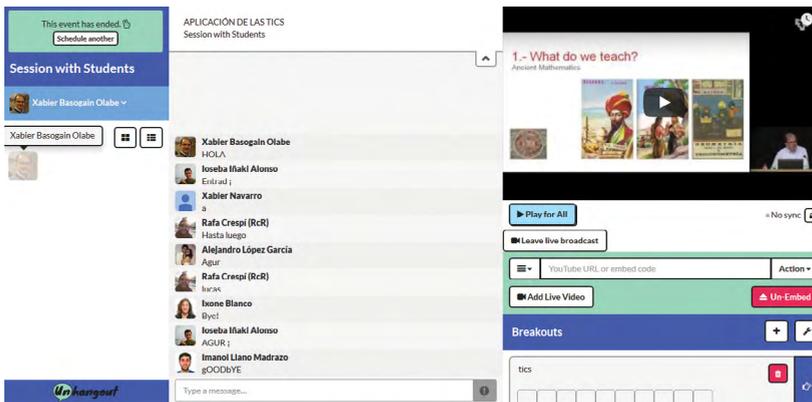
El tercer ejemplo corresponde al proyecto “Introducción del pensamiento computacional en colegios de Colombia”, desarrollado entre el Ministerio de Tecnologías de la Información y las Comunicaciones, la Universidad del País Vasco y la Red Nacional Académica de Tecnología

Avanzada, Renata, presentado en el stand del país en Virtual Educa 2017 (Basogain *et al.*, 2017).

Además, el equipo investigador divulga el uso de nuevas herramientas en línea que sirvan de catalizadores de comunicación entre alumnos, al realizar videoconferencias en las que se fomenta el intercambio de ideas, dudas, sugerencias, etc. Las herramientas elegidas son: 1) el entorno basado en Unhangout, una plataforma de código abierto para ejecutar en línea eventos a gran escala dirigidos por participantes; y 2) el sistema de código abierto de videoconferencia Mconf.org, que permite la comunicación de audio, video, proyectos y ficheros.

La figura 4 muestra la utilización de ambas plataformas para realizar un curso del máster de Ingeniería de Control, Automatización y Robótica de la Universidad del País Vasco, titulado “Aplicación de las TIC en la investigación”.

Figura 4. Utilización de las herramientas Unhangout y Mconf



Las herramientas síncronas son cada vez más accesibles y fáciles de incorporar a los procesos de enseñanza-aprendizaje. La motivación de los estudiantes aumenta a medida que se sienten acompañados en sus cursos presenciales o en línea.

Los sistemas de conferencias a través de la red se están incorporando a la enseñanza presencial y a distancia por el valor añadido que ofrecen a la asistencia del profesor o tutor en las diferentes modalidades de sesiones colaborativas síncronas (tutoría grupal o individual, grupos de trabajo o grabaciones de clases magistrales).

Líneas futuras de investigación

Otras líneas de investigación que estamos explorando en el grupo incluyen: 1) modelos cognitivos del proceso de aprendizaje, inteligencia, creatividad y lenguaje (Basogain, 2016a; 2016b), donde se enfatiza el potencial de Scratch y Snap para respaldar el PC en nuestros estudiantes y mejorar su efectividad en la resolución de problemas; 2) la incorporación de espacios para hacer en el ámbito educativo, lo que conlleva romper las barreras entre el aprendizaje formal e informal y combinar tanto el *software* como el *hardware* para aprender (Basogain *et al.*, 2018).

El equipo también está trabajando en la definición de una ontología para describir un proyecto STEAM en términos computacionales, denominada STEAM Computacional. Esta incluye los objetivos establecidos por la industria, el contenido destinado a su plan de estudios, la metodología para la enseñanza y el aprendizaje, y las herramientas de trabajo que los ciudadanos utilizarán en la sociedad contemporánea. Esta descripción formal de objetivos, contenido, métodos y medios está diseñada para servir como referencia estándar para el diálogo constructivo entre los constituyentes involucrados en el diseño de los proyectos STEAM: el sector académico, la industria y los responsables políticos nacionales.

STEAM Computacional se presenta como la combinación de tres componentes fundamentales: 1) contenido o currículo; 2) metodología de la enseñanza y el aprendizaje; y 3) la computación como lenguaje generador de la acción (Basogain *et al.*, 2018). Esta herramienta

responde a las necesidades de un contexto en el cual el PC sigue siendo un área de interés de estudio e investigación para su incorporación a la educación.

Además, en los últimos años, el PC ha sido integrado como herramienta tecnológica y lenguaje generativo matemático en el movimiento de la formación en STEAM.

Referencias

- Aprendiendo Scratch. (2015). Recuperado de <http://educacion-pyeduca.blogspot.com/2013/04/aprendiendo-scratch.html>
- Attwell, G. y Hughes, J. (2010). *Pedagogic approaches to using technology for learning: Literature review*. Recuperado de https://www.researchgate.net/publication/279510494_Pedagogic_approaches_to_using_technology_for_learning_literature_review
- Basogain, X., Olabe, M. A., y Olabe, J. C. (2015a). *Computación en la Educación a través de MOOC*. Curso “Pensamiento Computacional en la Escuela” Actas del VII Congreso Internacional de Computación y Telecomunicaciones, COMTEL 2015. 23-26 de Septiembre de 2015, Lima, Perú, 2015
- Basogain, X., Olabe, M. A., y Olabe, J. C. (2015b). *Curso Pensamiento Computacional en la Escuela: Diseño e Implementación en Miríada X Actas de las XXIII Jornadas Universitarias de Tecnología Educativa*. Badajoz, 11-12 de junio de 2015.
- Basogain, X., Olabe, M. A., y Olabe, J. C. (2016a). *PC-01: Introduction to computational thinking. Educational technology in primary and secondary education*. Conferencia presentada en XVIII Simposio Internacional de Informática Educativa.
- Basogain, X., Olabe, M. A., y Olabe, J. C. (2016b). *Pensamiento computacional utilizando tecnología educativa: aprendizaje mixto en las escuelas*. En XIX Congreso Internacional Edutec. EDUCACIÓN Y TECNOLOGÍA. Propuestas desde la investigación y la innovación educativa (pp. 441-443). Barcelona: Octaedro.
- Basogain, X., Olabe, M. A., Olabe, J. C., Rico, M., Rodríguez, L., y Amórtegui, M. (2017). *Pensamiento computacional en las escuelas de Colombia:*

- colaboración internacional de innovación en la educación*. Conferencia presentada en Virtual Educa Colombia.
- Basogain, X., Olabe, M. A., Olabe, J. C., y Rico, M. (2018). Computational thinking in pre-university blended learning classrooms. *Journal Computers in Human Behavior*, 80, 412-419.
- Basogain, X., Olabe, J. C., Olabe, M. A., Romero, A., Tejada, E., López de la Serna, A., Garay, U., Bilbao, N., y Castaño, C. (2019). *Design of collaborative computational ecosystems for the development of STEAM competencies in schools*. Conferencia presentada en 13th International Technology, Education and Development Conference.
- Benfield, G., Roberts, G., y Francis, R. (2006). *The undergraduate experience of blended e-learning: A review of UK literature and practice*. London: Higher Education Academy.
- Berkeley University. (2017). *The beauty and joy of computing*. Recuperado de <http://bjc.berkeley.edu/>
- Breslow, L., Pritchard, D. E., DeBoer, J., Stump, G. S., Ho, A. D., y Seaton, D. T. (2013). Studying learning in the worldwide classroom: Research into edX's first MOOC. *Research & Practice in Assessment*, 8, 13-25.
- Bruff, D. O., Fisher, D. H., McEwen, K. E., y Smith, B. E. (2013). Wrapping a MOOC: Student perceptions of an experiment in blended learning. *Journal of Online Learning and Teaching*, 9(2), 187.
- College Board. (2016). AP Computer Science Principles. Course and exam description. Recuperado de <https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf>
- Coursera. (2015). Code yourself! An introduction to programming. Universidad de Edimburgo, Universidad ORT Uruguay. Recuperado de <https://www.coursera.org/course/codeyourself>
- Department for Education England. (2013). National curriculum in England: Computing programs of study-key stages 1 and 2. Recuperado de <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study>
- Disessa, A. (2000). *Changing minds: Computers, learning, and literacy*. Cambridge: MIT Press.
- edX. (2015). *CS002x Programming in Scratch*. Harvey Mudd College. Recuperado de: <https://www.edx.org/course/programming-in-scratch>

- Glance, D. G., Forsey, M., y Riley, M. (2013). The pedagogical foundations of massive open online courses. *First Monday*, 18 (5). Recuperado de <http://firstmonday.org/ojs/index.php/fm/article/view/4350/3673>
- Johannesen, M. (2013). The role of virtual learning environments in a primary school context: An analysis of inscription of assessment practices. *British Journal of Educational Technology*, 44, 302-313. doi: 10.1111/j.1467-8535.2012.01296.x
- Kahneman, D. (2003). Maps of bounded rationality: Psychology for behavioral economics. *The American Economic Review*, 93(5), 1449-1475. doi: 10.1257/000282803322655392
- Kahneman, D. (2011). *Thinking, fast and slow*. New York: MaCmillan.
- Lu, J., y Law, N. W. Y. (2012). Understanding collaborative learning behavior from Moodle log data. *Interactive Learning Environments*, 20(5), 451-466.
- Mak, S., Williams, R., y Mackness, J. (2010). *Blogs and forums as communication and learning tools in a MOOC*. Conferencia presentada en 7th International Conference on Networked Learning, University of Lancaster, Lancaster.
- Misión Scratch. (2010, noviembre 18). *La xo en el aula*- I. E. Isabel la Católica [Entrada de blog]. Recuperado de <http://actividadesxo.blogspot.com.es/2010/11/visita-de-espana.html>
- Olabe, J. C., Basogain, X., y Olabe, M. A. (agosto, 2010). *Teaching and learning Scratch in schools around the world*. Conferencia presentada en Scratch@MIT. Massachusetts
- Olabe, J. C., Basogain, X., y Olabe, M. A. (2011). *Aprendiendo Scratch* (8 vol.). Ebook: Lulu. Recuperado de <http://www.lulu.com/shop/search.ep?keyWords=aprendiendoscratch&sorter=averageRating-desc>
- Olabe, J. C., Basogain, X., y Olabe, M. A. (agosto, 2015). *HelloScratchJr.org: Curricular design and assessment tools to foster the integration of ScratchJr and computational thinking into K-2 classrooms*. Conferencia presentada en 7th international Scratch conference. Scratch Creative communities. Proceedings of Scratch Conference, Ámsterdam, Holanda.
- Olabe, J. C., Basogain, X., y Olabe, M. A. (2016a). Developing new educational frontiers through breakthroughs in cognitive computation and new dimensions in pedagogical technology. *International Journal of Social Science and Humanity*, 6(11), 813-820. doi:10.18178/ijssh.2016.V6.755

- Olabe, J. C., Basogain, X., y Olabe, M. A. (2016b). Solving complex problems with a computational mind: An alternative to heuristic search. *International Journal of Learning and Teaching*, 2(1), 12-19.
- Olabe, J. C., Basogain, X., y Olabe, M. A. (2017). In search of a new space of K-12 mathematics in the classroom. *International Journal of Learning and Teaching*, 3(4), 328-335.
- Olabe, J. C., Basogain, X., y Olabe, M. A. (2019). An ontology of computational STEAM: The role of educational technology. En M. A. Clausen (ed.), *A closer look at educational technology. Technology in a globalizing world*. New York: Nova Science Publishers.
- Olabe, J. C., Basogain, X., Olabe, M. A., Maíz, I., y Castaño, C. (2014). Solving math and science problems in the real world with a computational mind. *Journal of New Approaches in Educational Research*, 3(2), 75-82. doi: 10.7821/naer.3.2.75-82
- Paddick, R. (2014, agosto 15). *As easy as VLE* [Entrada de blog]. Recuperado de http://edtechnology.co.uk/Article/as_easy_as_vle
- Papert, S. (1991). *Situating constructionism*. En I. Harel y S. Papert (eds.), *Constructionism* (pp. 1-11). Norwood, NJ: Ablex.
- Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning*, 1(1), 95-123.
- Pinker, S. (1995). *The language instinct: The new science of language and mind*. Inglaterra: Penguin.
- Pinker, S. (1999). *How the mind works*. Annals of the New York Academy of Sciences, 882(1), 119-127.
- Project Microworlds. (2017). Recuperado de <https://www.media.mit.edu/projects/microworlds/overview/>
- Proyecto PC-01 en Colombia. (2018). Recuperado de https://www.ehu.es/es/web/gmm/pc_colombia
- Proyecto PC-01 en República Dominicana. (2014). Recuperado de <https://www.ehu.es/es/web/gmm/minerd-pc01>
- Recursos en español. (2017). AprendiendoScratch. Recuperado de http://wiki.laptop.org/go/Recursos_en_espanol.
- Sancho Gil, J., y Padilla Petry, P. (2016). Promoting digital competence in secondary education: Are schools there? Insights from a case study. *Journal of New Approaches in Educational Research*, 5(1), 57-63. doi:10.7821/naer.2016.1.157

- Seliskar, H. V. (2014). *Using badges in the classroom to motivate learning*. *Faculty Focus*. Recuperado de <http://www.facultyfocus.com/articles/teaching-with-technology-articles/using-badges-classroom-motivate-learning/>
- Singh, J. (2015). *Learning analytics tools available in Moodle*. Recuperado de <http://www.moodleworld.com/learning-analytics-tools-available-in-moodle-moodleresearch-moodleworld/>
- Sykora, C. (2014, septiembre 11). *Computational thinking. Operational definition of computational thinking for K-12 education* [Entrada de blog]. Recuperado de <https://www.iste.org/explore/articleDetail?articleid=152>
- Universidad de Alicante. (2017). *Talleres en Moodle* [Entrada de blog]. Recuperado de <http://si.ua.es/es/moodle/actividades/talleres.html>
- Urtza Garay, E., Tejada, A., y López de la Serna, A. (2019). Learning in the 21st century for the inclusion of the makerspaces in universities. En M. A. Clausen (ed.), *A closer look at educational technology. Technology in a globalizing world*. New York: Nova Science Publishers.
- Vee, A. (2013). Understanding computer programming as a literacy. *Literacy in Composition Studies*, 1(2), 42-64.
- Vila, R. R., Mengual-Andrés, S., y Suárez, C. (2014). Evaluación de la calidad pedagógica de los MOOC. *Profesorado*, 18(1), 27-41.
- Wolfram, S. (2016, septiembre 7). *How to teach computational thinking* [Entrada de blog]. Recuperado de <http://blog.stephenwolfram.com/2016/09/how-to-teach-computational-thinking/>