Re Visión
Revista de Investigación en Docencia Universitaria de la Informática
AENUI Asociación de Enseñantes Universitarios de la Informática

Didáctica de la computación a través de la programación de ordenadores: un nuevo enfoque

Pablo Espeso,Carmen Hernández[1], Belén Palop[2]
CompuEdu @ UVa, Grupo de Computación Educativa de la Universidad de Valladolid
[1]Dpto. de Informática [2]Dpto. de Didáctica de las CC. Exp. Soc. y de la Matemática
Universidad de Valladolid

Resumen

Las etapas de educación Primaria y Secundaria son obligatorias en España, y deben proporcionar unas capacidades y destrezas básicas comunes a todos los niños preparándolos para ser «personas autónomas, críticas, con pensamiento propio». En la etapa de Secundaria, España ya se ha sumado de manera tímida al movimiento mundial para la introducción de la Computación en el ámbito escolar, que pronto llegará también a los currículos de Primaria. Para la correcta formulación de esta iniciativa, en este artículo presentamos nuestra visión e hipótesis de trabajo, adquirida y contrastada por nuestra amplia experiencia trabajando con niños a través de un modelo propio de metodología de enseñanza basado en el Aprendizaje Basado en Problemas, la Colaboración entre Pares y las Redes Personales de Aprendizaje, que creemos es extrapolable a las aulas formales. Desde nuestro punto de vista, es ahora el momento de reflexionar sobre la *Didáctica de la Computación*, una nueva disciplina científica que se ha de encargar de estudiar la relación entre los saberes, la enseñanza y el aprendizaje de las *Ciencias de la Computación*, y que comparte muchos aspectos con otras disciplinas afines como es, en particular, la Didáctica de las Matemáticas, cuyos avances y propuestas metodológicas elaboradas en las últimas décadas es imprescindible analizar y estudiar con profundidad.

Palabras clave: Alfabetización digital; Didáctica de la computación; Enseñanza primaria; Enseñanza secundaria; Lenguaje de programación.

1. Introducción

Fue en los años 60 del siglo XX cuando comenzó a crearse un movimiento en favor de la programación como herramienta educativa con el veterano Logo [4], que aún a día de hoy continúa utilizándose[1] para introducir a los más jóvenes al mundo de la programación y que ha sido el precursor de otros lenguajes más modernos, como por ejemplo Scratch, ya que este hereda de aquel algunas de sus características y funcionalidades.

En este artículo no vamos a debatir sobre las razones por las que la programación debe incorporarse a las actividades habituales del sector de la educación, en diferentes cursos y edades, pues sobre esto ya se ha discutido en multitud de ocasiones y existen numerosos estudios que indican que son muchos los beneficios obtenidos mediante el uso de estas herramientas [3, 5, 7]. De todos ellos el desarrollo del denominado Pensamiento Computacional [10] es el más destacado pero, desde nuestro punto de vista, una característica fundamental y poco analizada de las herramientas de programación actuales es su transversalidad, lo que permite que sean utilizadas en prácticamente cualquier área y materia del contexto escolar: matemáticas, lengua, música, dibujo…

Además, consideramos importante evidenciar el momento social en el que nos encontramos: inmersos en una era "tecnológica", es preciso e imprescindible infundir los conocimientos adecuados a los actuales escolares y futuros ciudadanos de la sociedad, para que puedan afrontar las problemáticas del futuro con las mejores habilidades posibles. De todas ellas, los conocimientos relativos a la tecnología ya están siendo esen-

[1]Véase por ejemplo la Turtle Academy: `https://turtleacademy.com/`

[2]Véase por ejemplo la web de ICT en educación de la Comisión Europea: `https://ec.europa.eu/digital-single-market/en/ict-education`

ciales y seguirán ganando relevancia en los próximos años[2].

Es ahora cuando debemos coger el tren que nos lleve a incorporar actividades específicas de programación en el aula o, de modo más general, actividades de computación. Este término es precisamente un elemento clave, ya que nos permite referirnos de un modo más amplio a lo que la tecnología puede aportar, que no es únicamente la programación sino el conocimiento, las habilidades y las capacidades que se adquieren a través de ella. No es programar por el hecho de aprender a programar, sino por lo que la programación, como herramienta, aporta dentro del aprendizaje de las ciencias de la computación y que va desde el manejo básico del ordenador hasta la comprensión más profunda y crítica del impacto de la tecnología en nuestras vidas.

Fue en 2009 cuando CompuEdu@UVa (`https://scratch.infor.uva.es/`) comenzó a trabajar en la formación de formadores desde el Máster de Formación de Profesor.[3] Iniciábamos por entonces el trabajo de reflexión sobre los currículos de tecnología y de informática, y sobre la enseñanza de la programación en particular. Desde aquellos años hasta la actualidad, hemos puesto en marcha multitud de iniciativas, tales como talleres y actividades para niños y jóvenes, cursos de formación de formadores o asistencia a congresos y foros de debate. Todo esto nos ha permitido comprobar el contexto en el que nos movemos en el presente, marcado por la relevancia social de la cuestión y por la necesidad de seguir trabajando y mejorando. Es evidente que así debe ser, atendiendo a estudios que afirman que buena parte de la población desconoce algunos conceptos esenciales de la computación [1].

Pero más allá de la difusión y divulgación en la sociedad, útil y necesaria, si pretendemos que la computación se incluya en el currículo de Primaria y Secundaria con ciertas garantías de calidad es fundamental el trabajo con los formadores, es decir, tanto los actuales docentes como los estudiantes de las Facultades de Educación. Apenas existen actividades, planes de estudio o asignaturas sobre tecnología ni sobre programación, y mucho menos auténticos especialistas en el terreno de la enseñanza de la computación. En España apenas comienzan a aparecer cursos de postgrado que trabajan estas competencias cuando, a nivel mundial, estos estudios están ya asentados.[4] Nosotros defendemos que unas competencias básicas en computación no deberían ser una especialidad de algunos docentes, sino una parte básica, común y obligatoria para cualquier maestro del S.XXI. Por todo esto, y atendiendo a las necesidades sociales para las próximas décadas, entendemos que es imprescindible trabajar en España en la *didáctica de la computación*, entendida como una disciplina científica cuyo objeto de estudio es la relación entre los saberes, la enseñanza y el aprendizaje de las ciencias de la computación.

2. Aprendizaje de la programación Basado en Problemas

Una de las metodologías más habituales y extendidas tanto en la enseñanza de la programación a niveles universitarios como en la enseñanza de las matemáticas a todos los niveles es la que en el lenguaje cotidiano llamamos “hacer problemas”. Dado que, en ocasiones, se confunde esta idea con la metodología de Aprendizaje Basado en Problemas, hemos resumido en la figura 1 los pasos de cada una de ellas.

El Aprendizaje Basado en Problemas (ABP a partir de ahora) ha recibido muchísima atención por parte de la comunidad educativa y, en particular, en el área de Didáctica de la Matemática. Cuando hablamos del ABP, nos referimos a un proceso en el que los estudiantes se mueven con ayuda del tutor en un ciclo de definición del problema, identificación de hechos relevantes, generación de hipótesis, identificación de necesidades cognitivas, aprendizaje autodirigido y abstracción del conocimiento generado hacia la solución del problema [8]. El ABP se orienta al aprendizaje significativo por parte del alumno y está basado en el constructivismo. En cuanto al tutor, según el concepto de Vygotsky [9] debe ser el docente el que facilite el aprendizaje, proporcionando un “andamio” que sujeta el aprendizaje únicamente cuando sea estrictamente necesario. Si bien la metodología del ABP no define en profundidad su imagen del estudiante, creemos conveniente indicar que los años de experiencia acumulada en niveles universitarios nos hacen apostar algo menos por el modelo Montessori de autoaprendizaje [6] e inclinarnos por el modelo de Vygotsky de coaprendizaje, considerando necesaria cierta instrucción o coinstrucción [9].

Desde que comenzamos nuestra labor como docentes en asignaturas de programación de ordenadores en la universidad y, más aún, desde que nos iniciamos en esta tarea para niveles preuniversitarios, hemos podido comprobar la potencia del ABP en la enseñanza de la programación. Si bien resulta difícil creer que un niño pueda, por sí mismo, descubrir conceptos tan abstractos como la modularidad o los bucles, hemos visto a lo largo de estos años cómo estos dos conceptos surgen de manera natural cuando utilizamos ABP con descubrimiento guiado por el docente. A modo de ejemplo, podemos relatar cómo un grupo de niños de segundo curso de primaria descubrió, autónomamente, los bucles en una actividad de programación “desenchufada”, de una hora de duración, como un recurso alternativo para no tener que volver a escribir una secuencia de instrucciones en un papel. De manera natural, en sus anotaciones aparecían comentarios como «2 veces», «hacer otra vez», o «bis». Para que esta necesidad surgiera, solo fue necesario que la secuencia que se debía repetir fue-

[3] Máster de Profesor de Educación Secundaria Obligatoria y Bachillerato, Formación Profesional y enseñanza de idiomas.

[4] Véanse por ejemplo las páginas web del máster en CS Education Computer Science Education de Stanford, `http://cs.stanford.edu/people/eroberts/mscsed/Admissions-MSInCSEducation.html`, Technology, innovation, and education en el Harvard Graduate School of Education, `https://www.gse.harvard.edu/masters/tie`, del programa de Computing in Education de la Universidad de Columbia, `http://www.tc.columbia.edu/mathematics-science-and-technology/communication-media-and-learning-technologies-design/degrees/computing-in-education-online-ma/`, y de Computing in Education del King’s College London, `http://www.kcl.ac.uk/study/postgraduate/taught-courses/computing-in-education-ma.aspx`

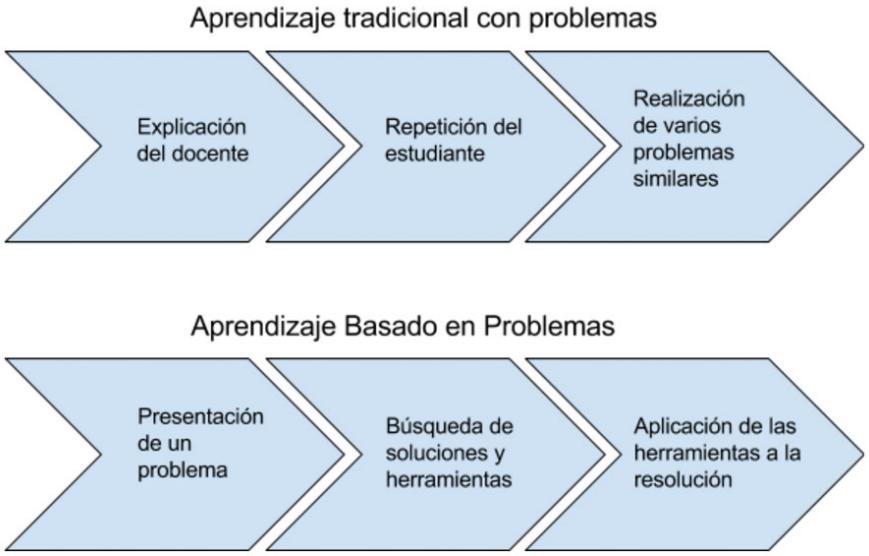


Figura 1: Aprendizaje tradicional con problemas contra Aprendizaje Basado en Problemas (ABP)

se lo suficientemente larga. Obviamente, no todos los niños la desarrollaron al tiempo, pero únicamente hubo que permitir el intercambio de información entre ellos para que el "invento" se extendiera como la pólvora. Volviendo a las fases de ABP de la figura 1, una vez provocada la necesidad de los bucles a través de un problema (Fase 1), los alumnos pudieron descubrir cierto conocimiento de manera autónoma (Fase 2), que aplicaron para la resolución del problema (Fase 3).

Como ya hemos dicho anteriormente, no creemos que todos los conceptos puedan llegar a desarrollarse de esta manera tan natural cuando trabajamos a niveles suficientemente elevados de complejidad y abstracción; de hecho, nunca hemos visto surgir los *arrays* o vectores como un autoaprendizaje en todos estos años. Es por ello que creemos que la aplicación del ABP al terreno de la computación se debe apoyar de manera fundamental en un tutor que coinstruye y que puede, en el momento adecuado, facilitar un descubrimiento más guiado, proporcionar una instrucción directa o trabajar en la práctica de una habilidad para fijarla por repetición. Esencialmente, lo que sugerimos es un enfoque mixto y equilibrado de autodescubrimiento a través del ABP y de instrucción y repetición. Llegar por uno mismo al concepto abstracto de bucle, descubrir su utilidad y hacer la idea propia, son una cosa. Ser capaz de resolver un problema con un bucle que recorra todos los elementos de una lista, requiere de otro tipo de competencias.

3. De la programación a la computación

La competencia digital [2], el pensamiento computacional y la programación de ordenadores se han mezclado en la literatura en los últimos años provocando quizás cierta confusión entre los diferentes términos, pero por cuestiones perfectamente comprensibles: la programación de ordenadores es una de las herramientas más habitualmente utilizadas para el desarrollo del pensamiento computacional que, además, trabaja de manera colateral competencias como la seguridad, la gestión de información o la comunicación.

Desde CompuEdu@UVa hemos trabajado con formadores de disciplinas tan afines como las matemáticas o la tecnología y tan dispares como la música o la lengua, siempre desde el convencimiento de que "programarlo" es una manera más de trabajar un determinado concepto. Introducir una breve partitura para que suene en un proyecto de Scratch provoca la profundización en el concepto de silencio o de duración de una nota. Desarrollar una trama entre unos personajes con Alice (`http://www.alice.org/index.php`) o con Ren'Py (`https://www.renpy.org/`) nos obliga a la escritura de diálogos y al desarrollo de una narrativa. Resolver cómo reaccionará la máquina ante las distintas respuestas de un usuario desarrolla competencias como la visión crítica de la tecnología y puede provocar un profundo debate sobre la misma en la sociedad actual. En esencia, la programación es una herramienta muy potente que podemos utilizar para completar los instrumentos pedagógicos de los que nos servimos en las aulas.

4. Caso de uso: el Club de Jóvenes Programadores

En 2013, y tras algunas pruebas piloto previas, pusimos en marcha el Club de Jóvenes Programadores de la Universidad de Valladolid (CJP@UVa, `http://scratch.infor.uva.es/cjp-uva`). Una actividad extraescolar, aún vigente, que

se realiza en la Escuela de Ingeniería Informática del Campus Miguel Delibes de Valladolid, y que fue ideada con el objetivo de promover el uso de la programación en jóvenes a partir de los 7 u 8 años.

Todos los lunes, a razón de tres horas semanales (de 17:00 a 20:00), un equipo de monitores especializados (de perfil técnico, generalmente Ingenieros Informáticos o estudiantes del respectivo grado a los que les proporcionamos formación específica sobre herramientas, metodologías y propuestas de programación en educación) actúan como guiadores y facilitadores para los alumnos, a través de una propuesta metodológica propia y diferente. A principios del año 2016 se inició la actividad con una filosofía similar en el campus de Segovia, también dependiente de la Universidad de Valladolid, haciendo que en total dispongamos de una muestra de participantes de unos 45 chavales con edades comprendidas entre 8 y 16 años (frente a los aproximadamente 15 jóvenes que estrenaron el Club en aquel 2013).

Como decíamos, nos apoyamos fundamentalmente en el Aprendizaje Basado en Problemas y aprovechamos que no debemos ceñirnos a un currículo concreto ni tampoco a libros, apuntes o una evaluación final. Gracias a esta libertad, hemos podido aprender durante estos años de unos alumnos de un perfil muy singular (que describimos más adelante).

De todos esos aprendizajes, hemos podido concretar una metodología orientada al aula tradicional y con un currículo más cerrado que es la que llevamos a cabo en las actividades extraescolares que también impartimos para colegios e institutos.

Habitualmente comenzamos utilizando la herramienta Scratch y, según el momento, el grupo o, incluso, el individuo, los introducimos en otras como Arduino, AppInventor, AppLab, Stencyl, Makey Makey, Ren'Py, etc. Para cada sesión, se plantean propuestas de enunciado muy abierto para que los participantes puedan adaptarlas a sus intereses, así como a su edad y experiencia con la programación. Esto nos permite que sean los propios chavales los que tengan el protagonismo en las creaciones: deciden qué programar y cómo hacerlo, afrontando —y resolviendo— los problemas, de modo que ellos son los líderes de su propio aprendizaje. Es importante resaltar que los Clubes alojan en un mismo grupo a personas con diferencias de hasta 6 años de edad, resultando en un intercambio muy rico de habilidades y en un entorno que facilita a los alumnos talentosos su implicación en niveles muy por encima de su edad cronológica.

Otro elemento clave del CJP@UVa es la disposición del mobiliario que facilita la colaboración entre pares en las tareas propuestas, así como el intercambio de otras experiencias, creando los asistentes una Red de Aprendizaje Personal. Algunos ejemplos de aprendizajes que se dan entre pares en el entorno del Club son la manera en que realizan cierta tarea en Minecraft o cómo crear un ejecutable `.bat`. Los Clubes facilitan y fomentan estos intercambios, tanto dentro como fuera de la actividad y evitando siempre la intervención de los facilitadores en las interacciones de aprendizaje entre pares. En Valladolid se dispone de grandes mesas con espacio para entre 4 y 6 alumnos, de modo que existe espacio para el debate y la conversación, para ver qué está haciendo el compañero y también para ayudarse los unos a los otros. En ocasiones se anejan varias de estas mesas para disponer de grupos de trabajo aún mayores. Por su parte, en Segovia, las mesas son individuales y están distribuidas en forma de U para facilitar el contacto entre todos los participantes.

La investigación pre-experimental

Dado que no existen unos objetivos cerrados de enseñanza ni tampoco unas competencias que debamos satisfacer de manera reglada en el transcurso del curso, el CJP@UVa es un espacio de aprendizaje e intercambio para los asistentes. Para nosotros es también una actividad en la que aprendemos constantemente sobre el proceso de enseñanza-aprendizaje de la programación. Gracias al CJP@UVa, hemos aprendido que los eventos son más intuitivos que las variables, o que no es sencillo realizar un descubrimiento autónomo de los *arrays*, pero sí de los sensores. O que las coordenadas cartesianas se pueden comprender y manejar con soltura ya en segundo curso de Primaria. De cara a formalizar algunos de estos aprendizajes, durante el curso 2014–15 se desarrolló una primera investigación con el objetivo de determinar las características principales del grupo de participantes que forman parte de la actividad. A partir de los resultados de la ejecución de dos tests psicométricos en tres fases (pretest, postest y seguimiento), se realizaron las mediciones de la evolución de las aptitudes mentales primarias y de la atención y concentración de los asistentes.

Las características más destacables de la muestra de participantes en esta primera investigación las resumimos en los siguientes puntos:

- Formaron parte 31 participantes, con edades comprendidas entre 7 y 15 años ($\bar{X} = 10{,}84$; $\sigma_X = 1{,}86$). De ellos, 24 son varones ($74{,}92\,\%$) y 7 mujeres ($25{,}08\,\%$).

- Los resultados fueron muy positivos en aptitudes tales como la concepción espacial y el razonamiento lógico, con percentiles de $82{,}58$ y $94{,}47$, respectivamente.

- Las variables en las que se observaron mayores incrementos entre la primera y la última fase de estudio fueron aquellas relacionadas con las habilidades lingüísticas: comprensión verbal y fluidez verbal, pasando de $56{,}59$ a $82{,}16$ y de $61{,}70$ a $85{,}00$, respectivamente.

- Las habilidades de atención y concentración se situaron en percentiles muy elevados en las tres fases, alcanzando percentiles medios de $96{,}11$ y $96{,}89$ respectivamente.

- Todas las variables objeto de estudio obtuvieron crecimientos positivos en cada nueva fase. En 5/7 de los casos, estos crecimientos fueron además estadísticamente

significativos a través de pruebas de estadística inferencial (test de Wilcoxon entre las dos fases extremas, pretest y seguimiento).

5. Conclusiones

La experiencia de todos estos años nos ha permitido aprender sobre el modo de enseñar a los jóvenes sobre computación en general, y más concretamente sobre programación. También hemos trabajado —y seguimos haciéndolo— en la continua mejora de nuestro modelo de enseñanza y en las propuestas pedagógicas con el objetivo de conseguir el mejor modo de introducción de la computación entre los jóvenes en edad escolar.

En este sentido, vemos muy necesario desarrollar y afianzar en España la *didáctica de la computación* como disciplina que debe apoyarse en la didáctica de las matemáticas como área afín, pero que debe abordar sus propios desafíos en cuanto al saber que la ocupa. Necesitamos profesionales que no solo sepan sobre computación, sino también cómo llevarla a las aulas. Estos profesionales deben conocer cuál es el momento del desarrollo más adecuado para la introducción de cada concepto y poder relacionarlo con el resto de aprendizajes escolares.

Lamentablemente, la adquisición de competencias digitales es una asignatura pendiente en la mayoría de facultades de Educación, por lo que aún estamos muy lejos de poder hablar de que sean estos los profesionales que se ocupen del desarrollo del pensamiento computacional en los niños. Si no actuamos con cuidado en este momento crucial, la programación de ordenadores entendida como un recetario de algoritmos a memorizar puede llegar a las aulas y ya sabemos, gracias a las pruebas internacionales sobre nuestro nivel de comprensión en matemáticas, que el giro del timón desde la memorización a la comprensión profunda y significativa no es sencillo. Debemos aprender mucho de cómo se enseña una disciplina afín, las matemáticas, y también de cómo no se debería enseñar, para no cometer los mismos errores.

Esta primera investigación acometida sobre el CJP@UVa nos ha permitido extraer las características de la muestra de participantes que forma parte de nuestro Club en Valladolid y observar que el aprendizaje de la programación con nuestra metodología mejora, de manera estadísticamente significativa, las capacidades y aptitudes de los individuos que hemos estudiado.

Sabemos que nuestra población es muy diferente a la que puebla habitualmente las aulas de Primaria y Secundaria, pero entendemos que hay elementos, tanto en las herramientas de programación que usamos, como en la propia metodología utilizada, que podemos extrapolar desde nuestros Clubes a las aulas ordinarias. Además, gracias a las experiencias adquiridas impartiendo actividades extraescolares, también estamos continuamente validando el tipo de problemas que mejor funcionan en ABP cuando se trabaja con una población promedio y cuya motivación, concentración y atención no responde a los elevados niveles de los Clubes.

Referencias

[1] FECYT, Google y Everis. *Educación en Ciencias de la Computación en España 2015*. Fundación Española para la Ciencia y la Tecnología (FECYT), 2016.

[2] Anusca Ferrari. Digcomp: A framework for developing and understanding digital competence in europe, 2013.

[3] G Fessakis, Evangelia Gouli y E Mavroudi. Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63:87–97, 2013.

[4] Wallace Feurzeig y S Papert. The logo programming language, 1967.

[5] Elizabeth Kazakoff y Marina Bers. Programming in a robotics context in the kindergarten classroom: The impact on sequencing skills. *Journal of Educational Multimedia and Hypermedia*, 21(4):371–391, 2012.

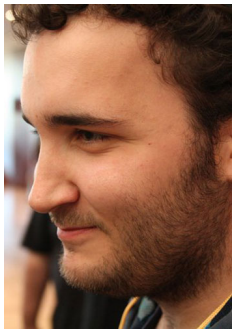
[6] Maria Montessori. *The absorbent mind*. Holt Paperbacks, 1949.

[7] Namje Park y Yeonghae Ko. Computer education's teaching-learning methods using educational programming language based on steam education. In *IFIP International Conference on Network and Parallel Computing*, pages 320–327. Springer, 2012.

[8] John R Savery. Overview of problem-based learning: Definitions and distinctions. *Interdisciplinary Journal of Problem-based Learning*, 1(1):3, 2006.

[9] Lev S Vygotsky. Mind in society: The development of higher mental process, 1978.

[10] Jeannette M Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, 2006.



Pablo Espeso Tascón es Ingeniero Informático, tiene un Máster en Investigación en Educación y colabora con medios digitales como Xataka o Educación 3.0, tratando temas relevantes sobre educación, tecnología y programación. Miembro del equipo CompuEdu@UVa, forma parte del equipo de monitores del CJP-Valladolid@UVa desde 2013 y es el coordinador de la actividad desde 2015.

Carmen Hernández Díez es profesora titular en el departamento de Informática (ATC, CCIA, LSI), en la Escuela de Ingeniería Informática de la Universidad de Valladolid. Ha participado en proyectos de innovación pedagógica en el ámbito de las Ingenierías. Pertenece a CompuEdu@UVa y desde febrero de 2013 colabora en el CJP-Valladolid@UVa, donde se fomenta la creatividad y el desarrollo del pensamiento computacional a niños de 8 a 14 años.

Belén Palop es Ingeniera en Informática y Doctora por la Universitat Politècnica de Catalunya. Tras de 15 años en el área de Lenguajes y Sistemas Informáticos, desde Septiembre de 2015 pertenece al área de Didáctica de la Matemática en la Universidad de Valladolid. Miembro del equipo CompuEdu@UVa, fundó el CJP-Valladolid@UVa en 2013 y el CJP-Segovia@UVa en 2016.