# Desarrollo y evaluación del pensamiento computacional: una propuesta metodológica y una herramienta de apoyo

## Development and assessment of computational thinking: a methodological proposal and a support tool

Alexis Daniel Fuentes Pérez, Gara Miranda Valladares
adfuentesp@gmail.com, gmiranda@ull.es

Departamento de Ingeniería Informática y de Sistemas
Universidad de La Laguna
San Cristóbal de La Laguna, Santa Cruz de Tenerife, España

***Resumen***- El objetivo de este trabajo ha sido la definición de una metodología para el desarrollo del pensamiento computacional y la posterior medición del desarrollo obtenido en el alumnado. La propuesta metodológica engloba tareas o actividades concretas para desarrollar el pensamiento computacional así como pruebas o tests que permitan analizar en qué medida se ha desarrollado el pensamiento computacional o las habilidades relacionadas con la resolución de problemas en un ámbito científico, específico, o incluso general o de aplicación práctica en situaciones de la vida cotidiana. Una vez definido el marco metodológico, se ha procedido a desarrollar una herramienta informática que proporciona un soporte global para la puesta en marcha de dicha metodología. La plataforma diseñada posee un diseño sencillo e intuitivo que permite un fácil uso de la misma ya que está dirigida a alumnado de enseñanza pre-universitaria y a profesorado que necesariamente no tiene por qué tener conocimientos avanzados de Informática. Por último, la plataforma proporciona, de forma automática, informes de progreso y desarrollo del curso entre los que se incluyen datos estadísticos por edades o género, entre otros.

***Palabras clave: pensamiento computacional, resolución de problemas, habilidades del siglo XXI, metodología para el desarrollo y la evaluación, educación pre-universitaria, plataforma web, Abstractly.***

***Abstract***- The objective of this work has been the definition of a methodology for the development of computational thinking and the subsequent measurement of the development obtained in the students. The methodological proposal encompasses specific tasks or activities to develop computational thinking as well as tests that allow analyzing the extent to which computer thinking or problem solving skills have been developed in a specific scientific or even practical situations of daily life. Once the methodological framework was defined, a web platform has been developed, thus providing a global support for the implementation of this methodology. The web platform has a simple and intuitive design that allows an easy use of it for pre-university students and teachers who do not necessarily have advanced computer skills. Finally, the platform provides – automatically - progress reports and course development including statistical data by age or gender, among others.

***Keywords: computational thinking, problem solving, 21st century skills, methodology for development and assessment, pre-university education, web platform, Abstractly.***

## 1. INTRODUCCIÓN

El pensamiento computacional es el razonamiento llevado a cabo en la formulación de un problema y en la expresión de su solución de forma que pueda ser llevado a cabo eficientemente por un agente que procesa información. Este término apareció por primera vez en 2006 cuando J. M. Wing publicó un artículo (Wing., 2006) en el que lo dio a conocer y explicó la manera de pensar de un científico en computación cuando se enfrenta a un problema. En este artículo también se expone la importancia que tendría la adquisición de esta *habilidad* en el resto de personas, ya que supondría una mejora en el procedimiento general de resolución de problemas que todos llevamos a cabo en nuestra vida cotidiana. Además, J. M. Wing ha demostrado un gran interés porque este nuevo concepto sea difundido en la educación, e invita al profesorado a que se involucre a motivar y a enseñar a sus alumnos en lo referente a las Ciencias de la Computación.

A partir de 2006, y tras esta primera referencia al término "*pensamiento computacional*" realizado por J. M. Wing, se ha comenzado a trabajar en este concepto, tratando de llegar a una definición operativa que describa con precisión sus características esenciales y ofrezca un marco de trabajo y un vocabulario común con el que los profesionales de la educación puedan trabajar. Quizás la iniciativa más conocida en este ámbito es la promovida por la Sociedad Internacional de la Tecnología en la Educación (ISTE) y la Asociación de Profesores de Informática (CSTA), quienes han definido formalmente el pensamiento computacional como el proceso de resolución de problemas que incluye las siguientes características:

- Formular problemas de forma que se permita el uso de un ordenador y otras herramientas para resolverlos.
- Organizar y analizar lógicamente la información.
- Representar la información a través de abstracciones como los modelos y las simulaciones.
- Automatizar soluciones haciendo uso del pensamiento algorítmico (estableciendo una serie de pasos ordenados para llegar a la solución).

- Identificar, analizar e implementar posibles soluciones con el objetivo de lograr la combinación más efectiva y eficiente de pasos y recursos.
- Generalizar y transferir este proceso de resolución de problemas para ser capaz de resolver una gran variedad de familias de problemas.

## 2. CONTEXTO

Partiendo de las definiciones anteriores y teniendo en cuenta que la forma natural en la que los Ingenieros Informáticos aprenden e interiorizan estos mecanismos de "pensamiento computacional" es mediante la programación informática o el diseño de *software* o aplicaciones informáticas, es normal que las principales líneas de trabajo en el ámbito del pensamiento computacional se hayan centrado principalmente en la enseñanza o difusión de la programación (DePryck, 2016; García-Peñalvo, 2016; García-Peñalvo, 2017) entre los más jóvenes y en la incorporación de especialistas en Informática en equipos de trabajo multi-disciplinares con el fin de que el conocimiento sobre programación y desarrollo de *software* que tienen los especialistas en Informática pueda ayudar en la resolución de problemas en otros ámbitos de conocimiento. Sin embargo, son prácticamente inexistentes los estudios que se han centrado en analizar cómo y en qué medida ayuda el pensamiento computacional a desarrollar las habilidades personales relacionadas con la resolución de problemas en general, la lógica o la creatividad (Grover & Pea, 2013). Es por ello que este trabajo se centrará en proporcionar una aplicación informática que proporcione soporte a la hora de evaluar o medir el desarrollo del pensamiento computacional en quienes aprenden programación y en qué medida este tipo de pensamiento está relacionado con el desarrollo de habilidades o competencias para la resolución de problemas.

Actualmente existen varias plataformas e iniciativas para fomentar el pensamiento computacional (Kelleher & Pausch, 2005; Lye & Koh, 2014), pero no se ha encontrado una que abarque todos los conceptos y requisitos necesarios que van desde la aplicación de una metodología, el desarrollo y seguimiento de actividades, la realización de mediciones de las capacidades desarrolladas en el alumnado, así como, la generación de informes sobre los progresos observados a lo largo del proceso. Es por ello que, previo al desarrollo de la herramienta informática que dé apoyo al proceso, ha sido necesario definir una metodología concreta que permita realizar un análisis cuantitativo sobre las ventajas de desarrollar el pensamiento computacional en el alumnado. De esta forma, la herramienta desarrollada permitirá poner en marcha en los centros de enseñanza pre-universitaria un proceso de desarrollo y análisis del pensamiento computacional que permitirá obtener resultados cuantitativos sobre las posibles ventajas que el pensamiento computacional ofrece en relación al desarrollo de otras habilidades cognitivas como pueden ser las vinculadas a la resolución de problemas en general. Con la puesta en práctica de esta metodología en distintos centros y niveles educativos podríamos obtener resultados que demuestren la importancia que el pensamiento computacional tiene en el desarrollo de habilidades cognitivas fundamentales para el alumnado del siglo XXI. Esto supondría un avance considerable a la hora de tratar de introducir el pensamiento computacional como eje de algunos currículos educativos, tal y como ya se ha hecho en algunos países.

## 3. DESCRIPCIÓN

Para el análisis a llevar a cabo, la metodología propuesta se basa en la realización de un pre-test y un post-test además de dividir al alumnado en dos grupos: un grupo experimental y un grupo de control. El alumnado del grupo experimental llevará a cabo actividades específicas para el desarrollo del pensamiento computacional. Por su parte, el alumnado del grupo de control realizará actividades totalmente diferentes y de ámbitos muy diversos y que nada tendrán que ver con el pensamiento computacional o la programación informática. Antes de comenzar con el desarrollo de las actividades seleccionadas, el alumnado (del grupo de control y del experimental) deberá llevar a cabo algún tipo de test o prueba que permita evaluar de alguna manera las habilidades cognitivas relacionadas con el pensamiento computacional o con la resolución de problemas en general (*pre-test*). A continuación, el alumnado llevará a cabo las actividades seleccionadas para cada grupo en cuestión. Al finalizar la formación, se volverá a llevar a cabo una medición de las habilidades cognitivas del alumnado (*post-test*), con el fin de tratar de observar alguna diferencia entre el progreso del alumnado del grupo de control y el del grupo experimental.

Uno de los temas más discutibles en el ámbito del pensamiento computacional es cómo se debe o se puede medir, pues no es algo trivial y no existe un método específico y aceptado para ello (Brennan, 2012; Román-González, 2015; Boix-Tormos, 2016). El pensamiento computacional es una habilidad muy relacionada con la lógica, la creatividad y la resolución de problemas, así que su medición podría realizarse de diferentes formas. La opción más extendida consiste en la realización de tests cuyas preguntas abarquen todas aquellas áreas de interés, pero también existen enfoques cuyos resultados son más cualitativos que cuantitativos, ya que se centran en observar cómo se abarca la resolución de un problema por un determinado individuo. En este trabajo, inicialmente se planteó la definición de un conjunto propio de pruebas (pre-tests y post-tests) para la evaluación, pero esta opción tiene la complejidad de tener que pasar por un proceso de correcta validación de los tests. Es por ello que como alternativa se decidió hacer uso de los tests definidos para los *concursos Bebras* (http://www.bebras.org), ya que están validados y presentan una gran variedad en cuanto a nivel de dificultades y tipos de problemas. Estos tests tienen duración límite de 40 minutos y están formados por un total de 18 preguntas, que se dividen en 6 de dificultad A (*fácil*), 6 de dificultad B (*media*), y 6 de dificultad C (*difícil*). Una vez calculada la puntuación total se normaliza sobre 10 y es la calificación final que se guarda en el registro de la actividad de medición. Si el alumno abandona el test durante su realización o sale del mismo, el test se enviará automáticamente con las respuestas que tenga marcadas en ese momento. Por último destacar que estas actividades no pueden ser realizadas ni accedidas por el profesorado para evitar que los alumnos puedan ser informados del contenido de los mismos o tengan una idea previa del estilo de preguntas.

La herramienta informática a desarrollar deberá tener en cuenta los requisitos siguientes:

- Cada curso está definido por una duración y una dificultad (dependiendo de la edad del alumnado).
- Dentro de cada curso, se separará al alumnado en dos grupos: un grupo experimental y un grupo de control.

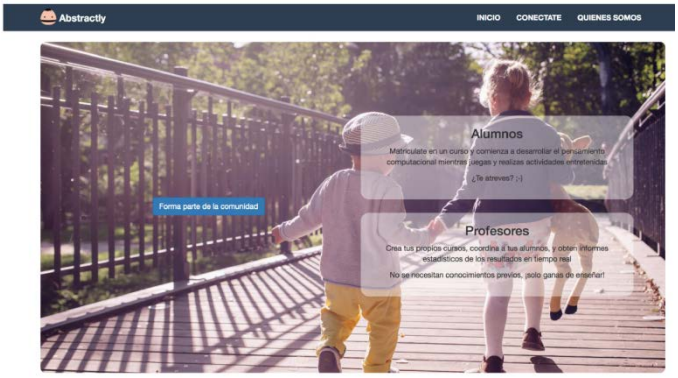
- Al comienzo de cada curso se realizará un test que evalúe el pensamiento computacional para tener una medida de las habilidades específicas del alumnado antes del inicio del curso en cuestión. Al finalizar el curso se realizará otro test, de características similares al realizado inicialmente. Estos test serán iguales para ambos grupos.
- Las actividades a realizar a lo largo del curso serán diferente para ambos grupos. El grupo experimental llevará a cabo actividades destinadas al desarrollo del pensamiento computacional, mientras que el grupo de control realizarán actividades de otro tipo. Para ambos casos, se ha recopilado un posible conjunto de actividades que, además han sido incorporadas en la plataforma. Para las actividades del grupo de control se proponen, entre otras, actividades relacionadas con la música, la creatividad, el deporte y con la revisión de contenidos de las asignaturas del currículo correspondiente. Para las actividades del grupo experimental se ha realizado un minucioso estudio de materiales disponibles en la literatura (Curzon & McOwan, 2017) y se ha optado por incluir en la plataforma una selección de actividades propuestas en proyectos e iniciativas como *Code.org*, *Teaching London Computing*, *CSunplugged*, *Google CS First*, *Codigo21*, *Google Computational Thinking for Educators*, *etc.* Las actividades incluidas son suficientes para proporcionar, de forma predeterminada, un curso completo para cada franja de edad. Sin embargo, la plataforma ofrece la posibilidad de que el profesorado pueda incluir tantas actividades como sea necesario.
- Cada curso tendrá actividades acorde a su dificultad (según rango de edad), aunque se ha introducido una adaptación progresiva de forma que al inicio del curso las actividades tendrán una dificultad menor, y al final del mismo se introducirán actividades de dificultad superior.
- La dificultad y el grado de adecuación de las actividades se configuran por defecto en el momento de su creación. Sin embargo, esta información se puede actualizar con la retroalimentación que proporcione el alumnado del curso.

## 4. RESULTADOS

*Abstractly*, nombre que toma la plataforma web, está destinada a promover y desarrollar el pensamiento computacional en edades tempranas dentro de las aulas. Proporciona al profesorado todas las herramientas necesarias para poder impartir un curso desde cero, sin necesidad de poseer conocimientos específicos, y además les permite gestionar tanto al alumnado del curso como a las actividades que realizan, así como llevar un seguimiento del progreso, y obtener informes estadísticos del progreso obtenido en cada momento del curso.

*Abstractly* ha sido desarrollada con *Ruby on Rails* principalmente, siguiendo las directrices del Modelo-Vista-Controlador (MVC), aunque algunas de sus funcionalidades han sido implementadas con *JavaScript*. Actualmente utiliza *PostgreSQL* como base de datos, y se encuentra desplegada en *Heroku*, una plataforma de computación en la nube, con un *Dynos* gratuito: http://abstractly.herokuapp.com/

La página principal de *Abstractly* presenta en unas pocas líneas la misión principal de esta plataforma e invita al usuario a participar en la experiencia, ya sea en el rol de estudiante o profesor. Podemos apreciar desde el inicio el diseño minimalista e intuitivo que encontraremos a lo largo de a la plataforma.




### *A. Creación de usuarios*

La creación de un usuario en la plataforma es bastante sencilla, consta de un formulario en el que se pide unos pocos datos, como su nombre completo, su fecha de nacimiento (la cual se puede introducir con un calendario interactivo), y su sexo (niño o niña). Por defecto, todos los usuarios obtienen el rol de usuario "básico" en el momento de crearse una cuenta. Existen diferentes maneras de cambiar este rol:

- *Rol "profesor"*. Los usuarios que quieran ser profesores deberán pedirlo a la plataforma y posteriormente les será concedido, cuando así sea el caso.
- *Rol de estudiante "experimental" o "control"*. Los otorga el profesor de un curso a sus alumnos matriculados en el mismo. Más adelante se explicará de forma detallada el proceso.

Destacar aquí que los permisos de administrador de la plataforma no cuentan como roles por motivos de seguridad, y estos solo pueden otorgarse desde el propio servidor. Además, esta característica permite a los propios administradores alternarse entre los cuatros roles diferentes que existen (básico, profesor, experimental, y control) sin perder sus permisos de administrador, con el fin de poder probar nuevas funcionalidades en fases de producción o permitir la detección y/o corrección de posibles bugs.

### *B. Perfiles*

Cada usuario tiene su propio perfil. En él se puede observar su información personal, como su nombre, edad, o rol actual en la plataforma. También se podrá ver su correo electrónico, si el usuario decide establecer, así como una imagen que es de carácter opcional. Destacar que también se puede observar un pequeño resumen de su actividad en *Abstractly*, y aquí se diferenciarían dos tipos de perfil diferentes:

- *Perfil estudiante.* Se muestra un listado de los cursos en los que ha estado, si realizó las pruebas de medición, y el porcentaje de actividades realizadas entre las totales de ese curso.

- *Perfil profesor*. Se muestra un listado de los cursos que ha dado, y la cantidad de total de alumnos que tuvo en cada uno de ellos.

### *C. Creación de cursos*

La creación de cursos está disponible para el profesorado desde la sección *"Mis Cursos"*. Un profesor puede crear y realizar distintos cursos al mismo tiempo con diferente alumnado. La duración del curso limitará la cantidad de actividades que se realizarán en el curso, y la dificultad (representada por el rango de edad del alumnado) fijará las pruebas de medición (pre-test y post-test) y el tipo de actividades a llevar a cabo durante el curso. Sin embargo, aunque las actividades a llevar a cabo durante el curso son propuestas por defecto cuando se crea un nuevo curso, éstas pueden ser eliminadas del curso, y ser sustituidas por otras a elección del profesor siempre que estas últimas no superen el tiempo total de duración estimado para el curso.

Una vez el profesor haya enviado los datos, se procederá a la creación del curso. *Abstractly* de forma automática rellenará el curso con las pruebas de medición y con las actividades de desarrollo correspondientes a los parámetros indicados. Las actividades de medición ya se encuentran asignadas a cada grupo de edad, y ya están conformadas por preguntas de diferente dificultad. También existe un conjunto de actividades de desarrollo predefinidas en la plataforma. En este caso, para incorporarlas a cada uno de los cursos, se aplicarán los siguientes criterios (el mismo criterio se sigue para ambos grupos, experimental y control):

*Curso: < 12 años*

- 70% de la duración: actividades con dificultad correspondiente a "< 12 años"
- 30% de la duración: actividades con dificultad correspondiente a "12 – 14 años"

*Curso: 12 – 14 años (14 – 16 años)*

- 20% de la duración: actividades con dificultad correspondiente a "< 12 años" ("12 – 14 años")
- 60% de la duración: actividades con dificultad correspondiente a "12 – 14 años" ("14 – 16 años")
- 20% de la duración: actividades con dificultad correspondiente a "14 – 16 años" ("16 – 18 años")

*Curso: 16 – 18 años*

- 30% de la duración: actividades con dificultad correspondiente a "< 12 años"
- 70% de la duración: actividades con dificultad correspondiente a "12 – 14 años"

### *D. Matriculación en cursos*

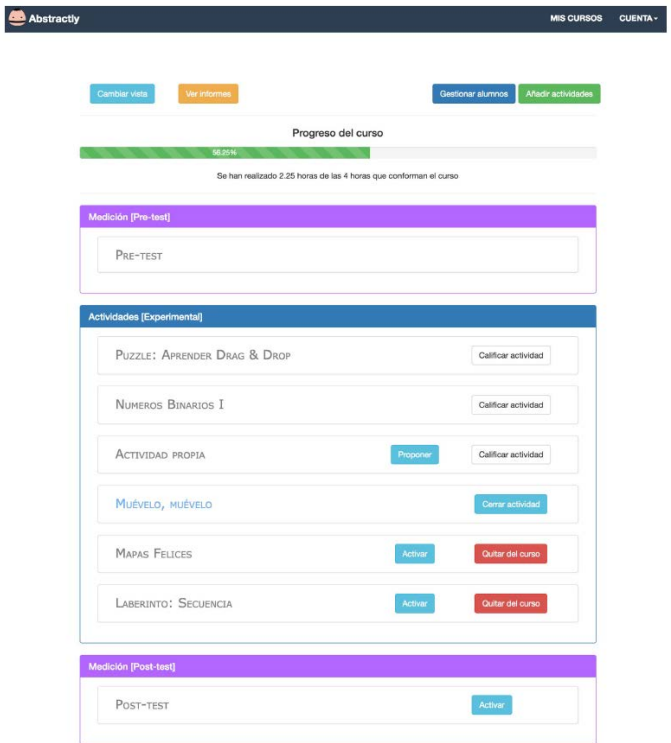
Para matricularse en un curso ofrecido en *Abstractly* basta con ir a la pestaña *"Mis Cursos"*. Ahí podrás encontrar un listado con aquellos cursos en los que el usuario está matriculado, así como acceder a la opción "Buscar Cursos". Desde esta opción aparecerá un listado con los cursos activos en este momento. Para matricularte en uno, basta con entrar en el que interesa e introducir la contraseña de matriculación. La contraseña de matriculación deberá ser proporcionada por el profesor a su alumnado. Un alumno no puede estar matriculado simultáneamente en más de un curso activo, puesto que esto "falsearía" los datos de medición que se obtendría en dichos cursos ya que dicho alumno estará realizando más actividades de las que están establecidas para su grupo correspondiente.

### *E. Vista del curso*

Los cursos son el objeto más importante dentro de la plataforma, ya que es a través de ellos donde se realiza la organización del alumnado, el desarrollo del pensamiento computacional, la obtención de informes, la organización de las actividades, etc. Es por esto que se ha querido que en la vista del curso, tanto el profesorado como el alumnado, tenga a su alcance todo lo necesario para el desarrollo y la medición a llevar a cabo. Al mismo tiempo que se muestra toda la información acerca de las actividades y el desarrollo del curso queremos que toda la interfaz y toda la información obtenida se muestre de forma sencilla e intuitiva.

En la parte superior del curso se encuentra un menú de acciones para el profesorado (únicamente visible y accesible para ellos):

- *Cambiar vista*. Permite cambiar la vista del curso entre "Grupo Experimental" y "Grupo de Control".
- *Ver informes*. Permite acceder a la generación automática de informes del curso.
- *Gestionar alumnos*. Permite administrar al alumnado matriculado, asignarlos a los grupos de control y experimental, desmatricularlos, etc.
- *Añadir actividades*. Abre la herramienta para añadir más actividades al curso, ya sean de *Abstractly* o se le da la opción al profesor de crear una propia con HTML o texto plano.



En la cabecera aparece el *"Progreso del curso"*, donde se puede apreciar para cada vista del curso el transcurso del mismo y permite observar si se va bien de tiempo, ya que no es lo mismo el tiempo estimado que se tiene para la realización de cada actividad, que el tiempo real que se emplea en realizarla. Ese tiempo lo introduce el profesorado con el registro de cada actividad, mientras que la plataforma se encarga de realizar los cálculos necesarios y actualizar esta barra de progreso.

A continuación, en la zona central de la vista de un curso aparecen los bloques de actividades del curso:

- *Medición [Pre-test]*. Actividad de medición que debe realizarse con la inicialización del curso. Permite obtener una medida de cómo están las capacidades de resolución de problemas y/o pensamiento computacional de cada alumno y del grupo antes de comenzar a desarrollarlas. Son iguales para ambos grupos (experimental y control).
- *Actividades.* Bloque de actividades para el grupo determinado (experimental o control, ya que son diferentes para cada grupo) incluidas dentro del curso. Por defecto están ordenadas por dificultad, pero se ha implementado un sistema de *drag-and-drop* para que el profesor pueda cambiar el orden a su gusto.
- *Medición [Post-test]*. Actividad de medición que debe realizarse con la finalización del curso. Permite obtener una medida de cómo están las capacidades de resolución de problemas y/o pensamiento computacional de cada alumno y del grupo al finalizar el estudio. Son iguales para ambos grupos (experimental y control).

Cada actividad dentro de un curso tiene un ciclo de vida, que es marcado por el profesor y se encuentra comprendido por:

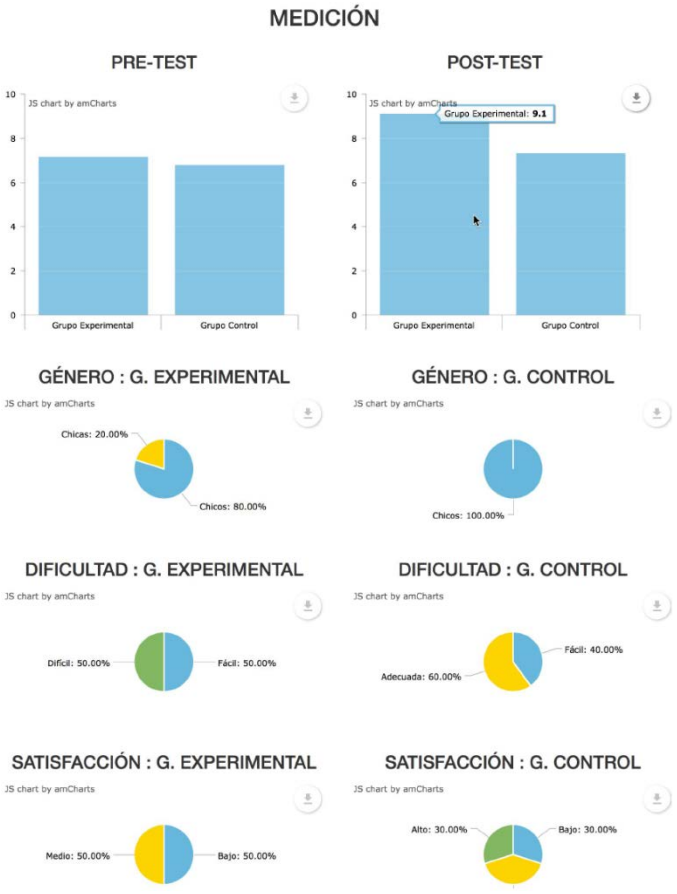
- *Fase 1 (En Espera).* La actividad está en el curso, pero no es accesible por el alumnado para su realización. En esta fase la actividad aún puede ser eliminada del curso por el profesor.
- *Fase 2 (Activada).* La actividad se encuentra activada para su realización. Se puede observar visualmente dicha condición, ya que se vuelve de un color azulado claro. Permite al alumnado entrar a la misma y registrar su realización.
- *Fase 3 (Finalizada).* La actividad se encuentra finalizada. El profesor deja su registro de la actividad, y establece la duración real que tuvo. El alumnado también registra una realimentación sobre la actividad. En concreto, puntúan la dificultad y el grado de satisfacción que han tenido con la actividad. Cada actividad puede obtener los valores de dificultad: *fácil*, *apropiada*, y *difícil*. El grado de satisfacción puede ser evaluado como: *bajo*, *medio*, y *alto*. Al finalizar, la actividad deja de ser accesible para el alumnado. Esto permitirá al profesor calificar la actividad a aquellos alumnos que registraron su realización. Visualmente la actividad deja de mostrarse como accesible. Además, si es una actividad propia, definida por el profesor y no de las predefinidas en la plataforma, se le permitirá proponerla para uso general.

*F. Generación de informes*

Uno de los puntos fuertes de la plataforma es la generación automática de informes para los cursos. Se ha utilizado la biblioteca externa de *amCharts* para facilitar la representación gráfica de los datos. Estos informes permiten obtener de una manera fácil y sencilla datos reales sobre el progreso del curso, como calificaciones, satisfacciones, comparaciones entre los diferentes grupos, etc. Estos datos son representados mediante gráficos y además de visualizarse en la plataforma, pueden descargarse (*.jpg, .png, .svg, .pdf*), exportarse (*.csv, .xls, .json.*) o imprimirse desde la propia plataforma.

A continuación veremos los diferentes tipos de informes que genera la plataforma:

- *Informes de medición.* Se realizan diferentes tipos de informes en relación a las calificaciones obtenidas en las actividades de medición (pre-test y post-test). En cada uno de los gráficos se muestra el resultado obtenido por el grupo experimental y el resultado obtenido por el grupo de control. En estos informes también se tienen en cuenta otras variables como son el género o la edad del alumnado.
- *Informes del curso.* Estos informes muestran información significativa del curso, como por ejemplo, porcentajes de alumnos de cada género por grupo, porcentajes de dificultad medios para cada actividad realizada en el curso, y porcentajes de satisfacción medios para cada actividad realizada en el curso.



## 5. CONCLUSIONES

Es sorprendente que siendo la Informática el gran motor de la innovación y el desarrollo tecnológico de la sociedad moderna, esta materia pase desapercibida en la mayor parte de los sistemas educativos actuales. Nadie pone en duda que los idiomas o las Matemáticas deban ser materias fundamentales en cualquier sistema educativo, pero son muy pocos los que han apostado por la Informática como elemento clave en la formación de los jóvenes. No es suficiente con introducir en los planes de estudios asignaturas puntuales dedicadas al conocimiento intrínseco de las tecnologías actuales y sus aplicaciones prácticas más inmediatas, pues esto sólo desarrolla en el alumnado destrezas para el manejo de un conjunto de herramientas concretas. Sería mucho más positivo, y enriquecedor para el alumnado, desarrollar destrezas para adecuarse a las nuevas tecnologías y herramientas que irán surgiendo y, ¿por qué no?, poder adquirir las habilidades suficientes para poder crear sus propias herramientas o llevar a la realidad sus propios proyectos tecnológicos.

Si estamos de acuerdo en que la Informática es clave para ofrecer soluciones a problemas abiertos en muchas disciplinas y no es una mera herramienta de "soporte" sino que juega un importante papel en la forma en que entendemos el mundo y los problemas que nos rodean, entonces la formación en este ámbito será primordial para que las generaciones futuras razonen computacionalmente, mejoren sus capacidades para la resolución de problemas y apliquen estas habilidades para transformar el mundo que nos rodea. Sin embargo, es difícil formar en este ámbito si no existe margen para ello en los correspondientes planes de estudio y si además, hay que luchar contra unos estereotipos preestablecidos y que deterioran la imagen de las Ciencias de la Computación y de quienes se dedican a ello. Incluso en ámbitos universitarios y en países que más concienciación existe al respecto, las titulaciones vinculadas con este tipo de formación parecen no tener excesiva acogida entre el alumnado. Según datos del proyecto Code.org actualmente en EE.UU. hay más de medio millón de puestos de trabajo vacantes en el sector de las Ciencias de la Computación pero apenas 43.000 estudiantes se graduaron el último año en titulaciones de este ámbito.

Es por todo ello que se considera primordial, que el primer paso en esta dirección esté centrado en la medición cuantitativa de las habilidades que el pensamiento computacional puede ayudar a desarrollar en nuestro alumnado más joven. Para ello, y con ánimo de simplificar esta tarea, se ha diseñado una herramienta informática que proporciona una propuesta metodológica para realizar dicho estudio. *Abstractly* es una plataforma que presenta todos los conceptos necesarios para desarrollar y medir el pensamiento computacional y que ha sido desarrollada y pensada exclusivamente para ello, por lo que ofrece un servicio, un producto y una dedicación muy concreta para este fin. Una de las cosas más importantes a destacar es que esta plataforma está preparada para que cualquier persona, docente o no, pueda llegar a impartir un curso sin necesidad de poseer conocimientos algunos en la materia, sólo ganas de enseñar y fomentar este tipo de habilidades. Otro de sus fuertes, su diseño intuitivo y simple, permite complementar este deseo de fomentar del pensamiento computacional entre joven alumnado en niveles pre-universitarios.

Como trabajo futuro, el siguiente paso reside en la puesta en marcha masiva de la herramienta en diferentes entornos educativos y grupos de edad, para así recopilar suficientes datos y evidencias estadísticas que nos permitan evaluar las ventajas que el pensamiento computacional puede incorporar en la educación de nuestros jóvenes y futuros ciudadanos del siglo XXI. Además, esta experiencia permitiría obtener una retroalimentación global y detallada que permitiría completar el conjunto de actividades ofrecidas por la herramienta así como mejorar algunas de sus características.

## AGRADECIMIENTOS

Parte de este trabajo ha sido financiado mediante una Beca de Colaboración del Ministerio de Educación, Cultura y Deporte.

## REFERENCIAS

Boix-Tormos, J. J. (2016). Estudio de la influencia del aprendizaje del pensamiento computacional en las materias de ciencias en alumnos de secundaria. Tesis Doctoral. Universitat Oberta de Catalunya.

Brennan, K. & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In Annual American Educational Research Association Meeting. Vancouver, Canada.

Curzon, P. & McOwan, P.W. (2017). The Power of Computational Thinking: Games, Magic and Puzzles to Help You Become a Computational Thinker. World Scientific.

DePryck, K. (2016). From computational thinking to coding and back. Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality (pp. 27-29). Salamanca, Spain: ACM.

García-Peñalvo, F. J. (2016). Proyecto TACCLE3 – Coding. In F. J. García-Peñalvo & J. A. Mendes (Eds.), XVIII Simposio Internacional de Informática Educativa, SIIE 2016 (pp. 187-189). Salamanca, España: Ediciones Universidad de Salamanca.

García-Peñalvo, F. J., Llorens Largo, F., Molero Prieto, X., & Vendrell Vidal, E. (2017). Educación en Informática sub 18 (EI<18). ReVisión, 10(2), 13-18.

Grover, S. & Pea, R. (2013). Computational Thinking in K–12 A Review of the State of the Field. Educ. Res., vol. 42, no. 1, pp. 38–43.

Kelleher, C. & Pausch, R. (2005). Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers. ACM Comput Surv., vol. 37, no. 2, pp. 83–137.

Lye, S.Y. & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? Comput. Hum. Behav., vol. 41, pp. 51–61.

Román-González, M. (2015). Computational Thinking Test: Design Guidelines and Content Validation. EDULEARN Conference. Barcelona.

Wing., J. M. (March de 2006). Computational Thinking. Communications of the ACM, 49(3).