

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/292398919>

# Test de Pensamiento Computacional: diseño y psicometría general [Computational Thinking Test: design & general psychometry]

CONFERENCE PAPER · OCTOBER 2015

---

READS

32

3 AUTHORS, INCLUDING:



[Marcos Román](#)

National Distance Education University

12 PUBLICATIONS 9 CITATIONS

[SEE PROFILE](#)



[Juan-Carlos Pérez-González](#)

National Distance Education University (UN...)

23 PUBLICATIONS 277 CITATIONS

[SEE PROFILE](#)

# Test de Pensamiento Computacional: diseño y psicometría general

## Computational Thinking Test: design & general psychometry

Marcos Román-Gonzalez<sup>1</sup>, Juan Carlos Pérez-González<sup>2</sup>, Carmen Jiménez-Fernández<sup>1</sup>  
mroman@edu.uned.es, jcperez@edu.uned.es, mjimenez@edu.uned.es

<sup>1</sup>Departamento MIDE I  
UNED  
Madrid, España

<sup>2</sup>Departamento MIDE II  
UNED  
Madrid, España

**Resumen-** El pensamiento computacional (PC) se viene situando en el foco de la innovación educativa como un conjunto de habilidades de solución de problemas que debe ser adquirido por las nuevas generaciones de estudiantes. Sin embargo, no existe aún consenso internacional sobre cómo definir el PC, ni tampoco una idea clara sobre cómo incorporarlo a los sistemas educativos en sus distintas etapas. Igualmente, hay un enorme vacío sobre cómo medir y evaluar el PC. En respuesta, presentamos el diseño de un *Test de Pensamiento Computacional* dirigido a estudiantes españoles de entre 12 y 13 años (1º y 2º ESO): describimos los principios bajo los cuales ha sido construido el test, así como su comportamiento psicométrico general a partir de una aplicación del mismo sobre una muestra de 400 sujetos. Finalmente, se discuten posibles usos y limitaciones de nuestro test, y sus posibles concurrencias con otras pruebas internacionales de evaluación del pensamiento computacional.

**Palabras clave:** *pensamiento computacional, test de pensamiento computacional, codigoalfabetización, educación en ciencias de la computación.*

**Abstract-** Computational thinking (CT) is being located as focus of educational innovation as a set of problem solving skills to be acquired by new generations of students. However, we still lack international consensus on a definition of CT, nor a clear idea of how to incorporate CT to our education systems at various levels. Similarly, there is a striking gap about how to measure and assess CT. In reply, this paper presents the design of a *Computational Thinking Test* aimed at Spanish students between 12 and 13 years old (grades K-7 & K-8): we describe the guidelines on which our test has been built and its general psychometric properties from an application on a sample of 400 subjects. Finally, possible uses and limitations of the test and possible concurrency of the same with other international evidence on computational thinking assessment are discussed.

**Keywords:** *computational thinking, computational thinking test, code literacy, computer science education.*

### 1. INTRODUCCIÓN

Vivimos inmersos en un ecosistema digital lleno de objetos programables controlados por software (Manovich, 2013). En este contexto, ser capaz de manejar el lenguaje de las computadoras emerge como una habilidad indispensable, un nuevo alfabetismo que nos permite participar de manera plena y efectiva en la realidad digital que nos rodea: se trata de ‘programar o ser programado’ (Rushkoff, 2010); de ser ‘app capacitado o app dependiente’ (Gardner, 2014). El término

‘codigoalfabetización’ (del inglés ‘code-literacy’) ha sido acuñado recientemente para referirse al proceso de enseñanza-aprendizaje de la lectoescritura con los lenguajes informáticos de programación (Román, 2014). Así, se considera que una persona está codigoalfabetizada cuando es capaz de leer y escribir en el lenguaje de los ordenadores y otras máquinas, y de pensar computacionalmente (Belshaw, 2013). Si la codigoalfabetización alude en última instancia a una nueva práctica de lectoescritura, el pensamiento computacional (PC) se refiere al proceso cognitivo subyacente de resolución de problemas que le da soporte.

En esta nueva realidad invadida por lo digital, no es sorprendente que haya un renovado interés en muchos países por introducir el PC como un conjunto de habilidades de solución de problemas que debe ser adquirido por la nueva generación de estudiantes; aún más, el PC se está empezando a considerar como un elemento central de todas las disciplinas STEM: acrónimo de ‘science, technology, engineering & mathematics’, es decir ‘ciencia, tecnología, ingeniería y matemáticas’ (Henderson, Cortina & Wing, 2007). Aunque aprender a pensar computacionalmente viene siendo reconocido como relevante desde hace largo tiempo (Papert, 1980), en la medida que la programación y la computación se han convertido en omnipresentes, acciones que sustentan la comunicación, la ciencia, la cultura y los negocios en nuestra sociedad digital (Howland & Good, 2015); el PC es progresivamente visto como una habilidad esencial que nos posibilita crear, en vez de sólo consumir, tecnología (Resnick et al., 2009).

Así, numerosos gobiernos alrededor del mundo están incorporando la programación informática en sus currículos educativos nacionales. Desde la propia Comisión Europea se están promoviendo acciones en esa misma línea para asegurar la competitividad de las futuras economías de la Unión; y, de hecho, 9 países europeos ya han incluido el ‘coding’ en sus centros educativos en menor o mayor medida: Bulgaria, Chipre, Dinamarca, Estonia, Grecia, Irlanda, Polonia, Portugal y Reino Unido (European Schoolnet, 2014). La reciente decisión de introducir las Ciencias de la Computación (‘Computer Science’) como materia troncal en todas las etapas obligatorias del sistema educativo de Reino Unido, es el reflejo más paradigmático del reconocimiento creciente de la importancia del PC en la formación de las nuevas cohortes de trabajadores y ciudadanos europeos (Brown et al., 2013).

## 2. CONTEXTO

Sin embargo, todavía hay poco consenso sobre una definición formal del PC, y discrepancias sobre cómo debería ser integrado en los currículos educativos (Gouws, Bradshaw & Wentworth, 2013a). Hace casi diez años, en 2006, Jeanette Wing publicó el artículo fundacional de la disciplina, en el que se definía que el PC “implica la resolución de problemas, el diseño de sistemas, y la comprensión de la conducta humana, haciendo uso de los conceptos fundamentales de la informática” (Wing, 2006, p. 33). La esencia del PC es pensar como un científico de la computación cuando uno se enfrenta a un problema. Pero esta primera definición genérica viene siendo revisada y especificada en intentos sucesivos a lo largo de los últimos años, sin llegar aún a un acuerdo generalizado sobre la misma.

Posteriormente, en 2008 Wing clarificó, “el pensamiento computacional incluye los procesos de pensamiento implicados en la formulación de problemas y de sus soluciones, de tal modo que éstos estén representados de una manera que pueda ser abordada efectivamente por un agente-procesador de información” (Wing, 2008, p. 3718), como un ordenador. Cuatro años más tarde, esta definición es simplificada por Aho, que declara el PC como el proceso de pensamiento involucrado en la formulación de problemas de tal manera que sus soluciones puedan ser representadas como pasos computacionales discretos y algoritmos (Aho, 2012). Por otra parte, en 2011 la ‘*Computer Science Teachers Association (CSTA)*’ y la ‘*International Society for Technology in Education (ISTE)*’ de los Estados Unidos, desarrollaron una definición operativa del PC que sirve de marco de trabajo y vocabulario compartido para los profesores de informática (‘*Computer Science Teachers*’) en las etapas de educación secundaria y preuniversitaria estadounidense. Siguiendo a la CSTA & ISTE (2011), el pensamiento computacional es un proceso de solución de problemas que incluye (aunque no está limitado a) las siguientes características:

- Formular problemas de un modo que se haga posible utilizar un ordenador y otras máquinas en su resolución.
- Organizar lógicamente y analizar datos
- Representar datos a través de abstracciones tales como modelos y simulaciones
- Automatizar soluciones a través del pensamiento algorítmico (una serie de pasos discretos y ordenados)
- 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 solución de problemas a una amplia variedad de situaciones.

Incluso Google ha aportado su propia definición de PC como “un conjunto de habilidades y técnicas de solución de problemas, que los ingenieros de software usan para escribir los programas informáticos que subyacen a las aplicaciones que usamos a diario (...) las 4 fases específicas del PC son: Descomposición de un problema o tarea en pasos discretos; Reconocimiento de patrones (regularidades); Generalización de dichos patrones y abstracción (descubrir las leyes o principios que causan dichos patrones); y Diseño algorítmico (desarrollar instrucciones precisas para resolver el problema y sus análogos)” (Google for Education, 2015, en línea).

A pesar de las definiciones formuladas anteriormente, se reconoce que aún no existe una idea clara sobre cómo incorporar el PC a los sistemas educativos en sus distintas etapas; existiendo una enorme variedad y heterogeneidad de intervenciones educativas al respecto (Lye & Koh, 2014). Igualmente, hay un enorme vacío sobre cómo medir y evaluar el PC, hecho que debe ser abordado. Y es que sin la atención suficiente sobre la medida y evaluación del PC, será muy difícil que éste se abra camino exitosamente en cualquier currículum. Aún más, para poder juzgar la efectividad de cualquier currículum que incorpore el PC, es necesario validar previamente instrumentos de medida que permitan a los educadores evaluar en qué grado los estudiantes han desarrollado esa habilidad de pensamiento (Grover & Pea, 2013).

La investigación más reciente que aborda la problemática de la medida y evaluación del PC, como el instrumento ‘*Fairy Assessment in Alice*’, se ha apoyado tanto en valorar objetos creados-programados por los estudiantes como en analizar sus respuestas a reactivos de programación prediseñados; en ambos casos, para tratar de medir la comprensión y uso de los estudiantes de la abstracción, la lógica condicional, el pensamiento algorítmico y otros conceptos del PC que se aplican para resolver problemas (Werner, Denner, Campe & Kawamoto, 2012). En Sudáfrica, se ha desarrollado el ‘*Computational Thinking Framework (CTF)*’, un marco de trabajo que sirve de referencia para diseñar y evaluar materiales y recursos de aprendizaje del PC (como, por ejemplo, Light-Bot), y evaluar a los estudiantes (Gouws, Bradshaw & Wentworth, 2013a). Este grupo de investigación ha estudiado igualmente el rendimiento de estudiantes de primer año de universidad en su ‘*Test for Computational Thinking*’ (Gouws, Bradshaw & Wentworth, 2013b): las preguntas usadas en esta evaluación fueron seleccionadas de entre el banco de ítems liberados por la ‘*Computer Olympiad Talent Search*’ (<http://www.olympiad.org.za/talent-search/>), una olimpiada informática cuyo objetivo es detectar y premiar el talento computacional, y que pretende orientar a los estudiantes sudafricanos con más aptitudes en PC hacia carreras técnicas.

En la Universidad de Kentucky, se han examinado las conexiones entre el PC y el pensamiento crítico en estudiantes universitarios, desarrollando su propio instrumento de medida del PC consistente en una combinación de preguntas cortas y de elección múltiple que trata de evaluar a los estudiantes en diversos conceptos computacionales; este instrumento aún no está validado ni resulta suficientemente fiable (Walden, Doyle, Garns & Hart, 2013). En Taiwán, recientemente se ha realizado una evaluación del PC a gran escala entre alumnos de secundaria, bachillerato y formación profesional (Lee, Lin & Lin, 2014), usando para ello una selección de 15 tareas-problemas extraídos del ‘*International Bebras Contest*’ (<http://www.bebas.org/>): un concurso a nivel internacional sobre fluidez informática y computacional en edades escolares.

Pero aún existe una ausencia notable de tests diseñados para la etapa de educación secundaria que hayan sido sometidos a un proceso completo de validación. Para contribuir a paliar este vacío, en este artículo se presenta el diseño del ‘*Test de Pensamiento Computacional*’ dirigido a estudiantes españoles de entre 12 y 13 años (1º y 2º ESO): describiremos los

principios bajo los cuales ha sido construido el test en su conjunto y cada uno de sus ítems, así como su comportamiento psicométrico general a partir de una aplicación del mismo sobre una muestra de 400 sujetos.

### 3. DESCRIPCIÓN

Nuestro ‘Test de Pensamiento Computacional’ (a partir de ahora, TPC) fue diseñado inicialmente (versión 1.0, de octubre de 2014) con una longitud de 40 ítems de elección múltiple. Tras un proceso de validación de contenido a través del procedimiento ‘juicio de expertos’, dicha versión inicial fue depurada a una versión final de 28 ítems de longitud (Román, 2015a); que está construida bajo los siguientes principios:

- **Objetivo:** el TPC pretende medir el nivel de aptitud-desarrollo del pensamiento computacional en el sujeto.
- **Definición operativa del constructo medido:** el pensamiento computacional es la capacidad de formular y solucionar problemas apoyándose en los conceptos fundamentales de la computación, y usando la lógica-sintaxis de los lenguajes informáticos de programación: secuencias básicas, bucles, iteraciones, condicionales, funciones y variables.
- **Población objetivo:** dirigido a población escolar española de entre 12 y 13 años (1º y 2º ESO).
- **Tipo de instrumento:** prueba objetiva de elección múltiple con 4 opciones de respuesta (sólo 1 correcta).
- **Longitud:** 28 ítems.
- **Tiempo máximo de realización:** 45 minutos.

Cada uno de los ítems está diseñado y caracterizado en las siguientes cinco dimensiones:

- **Concepto computacional abordado:** cada ítem aborda uno o más de los siguientes 7 conceptos computacionales, ordenados en dificultad creciente: Direcciones básicas (4 ítems); Bucles-‘repetir veces’ (4 ítems); Bucles-‘repetir hasta’ (4 ítems); Condicional simple-‘if’ (4 ítems); Condicional compuesto-‘if/else’ (4 ítems); Mientras que-‘while’ (4 ítems); Funciones simples (4 ítems). Estos conceptos están alineados con los estándares que fija la CSTA para la educación en ciencias de la computación en estas edades (CSTA, 2011).
- **Entorno-Interfaz del ítem:** los ítems del TPC se presentan en alguno de los siguientes dos entornos gráficos o interfaces: ‘El Laberinto’ (23 ítems); ‘El Lienzo’ (5 ítems).
- **Estilo de las alternativas de respuesta:** en cada ítem, las alternativas de respuesta se pueden presentar en alguno de estos dos estilos: ‘Visual por flechas’ (8 ítems); ‘Visual por bloques’ (20 ítems).
- **Existencia o inexistencia de anidamiento:** dependiendo de si la solución del ítem involucra una secuencia de comandos-órdenes con (19 ítems) o sin (9 ítems) conceptos computacionales anidados (un concepto embebido en otro concepto en un orden de jerarquía superior).
- **Tarea requerida:** dependiendo de cuál de las siguientes tareas cognitivas es necesaria para la resolución del ítem: ‘Secuenciación’, enunciar de manera ordenada una serie

de comandos-órdenes (14 ítems); ‘Completamiento’, completar un conjunto incompleto de comandos previamente dado (9 ítems); ‘Depuración’, depurar (‘debug’) un conjunto incorrecto de comandos previamente dado (5 ítems).

En las siguientes **Figuras 1, 2 y 3** se pueden ver ejemplos de algunos de los ítems diseñados, detallándose sus especificaciones en las distintas dimensiones.

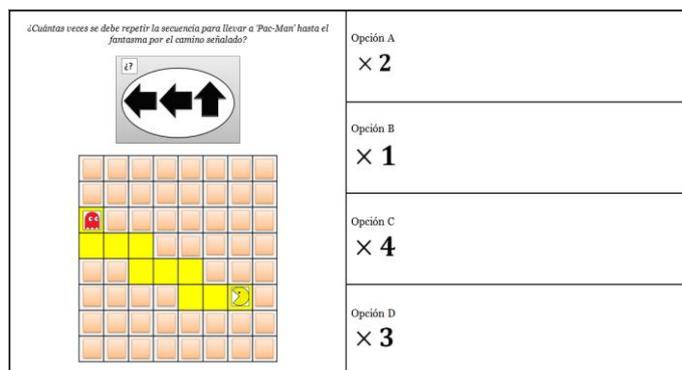


Figura 1. Ítem 6 del TPC: Bucle-‘repetir veces’; ‘Laberinto’; ‘Visual por flechas’; Sin anidamiento; ‘Completamiento’.

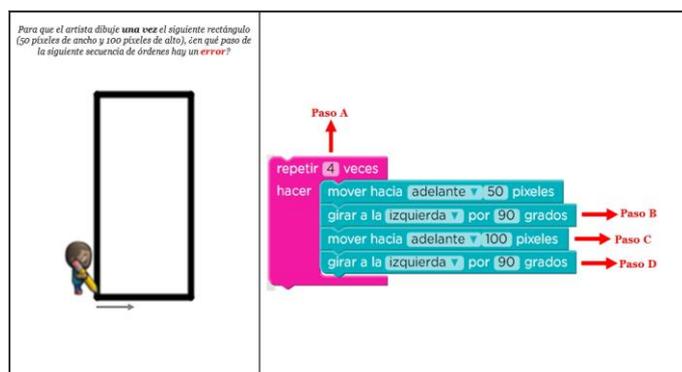


Figura 2. Ítem 7 del TPC: Bucle-‘repetir veces’; ‘Lienzo’; ‘Visual por bloques’; Sin anidamiento; ‘Depuración’.

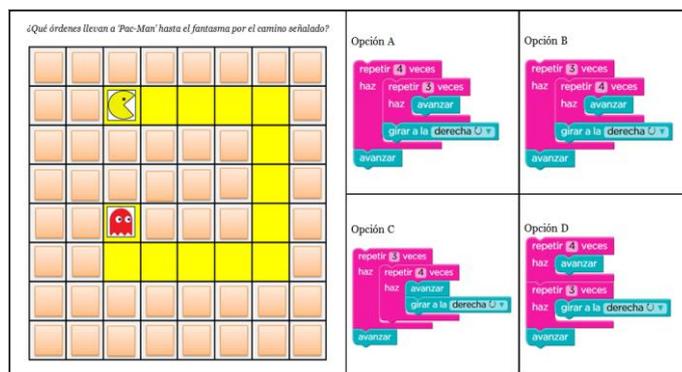


Figura 3. Ítem 8 del TPC: Bucle-‘repetir veces’; ‘Laberinto’; ‘Visual por bloques’; Con anidamiento; ‘Secuenciación’.

El TPC está siendo actualmente aplicado en amplias muestras de la población objetivo a través de un formulario en línea (Román, 2015b), programado con la tecnología de Google Drive Forms; y que puede navegarse a través del siguiente enlace: <http://goo.gl/IYEKMB>

#### 4. RESULTADOS

Se presentan unos análisis generales del comportamiento psicométrico de nuestro TPC a partir de su reciente aplicación sobre una muestra de 400 sujetos ( $n=400$ ) de la población objetivo. Las características de la muestra en lo relativo a las variables 'sexo' y 'curso' se ilustran en la **Tabla 1**.

Tabla 1. Características de la muestra.

		Curso		Total
		1º ESO	2º ESO	
Sexo	Chico	170	91	261
	Chica	105	34	139
Total		275	125	400

El procedimiento de muestreo utilizado es intencional y no probabilístico, dado que el instrumento es aplicado en los centros que posteriormente van a seguir el programa-curso 'K8 Intro to Computer Science' (Code.org, 2015), sirviendo así de prueba pre-test para la evaluación de dicho programa. Los centros participantes son 9 institutos de las tres provincias de la Comunidad Valenciana, que están previamente integrados en la red de innovación 'IT Teaching' cuyo objetivo es la enseñanza en inglés de la asignatura de informática; esta asignatura es optativa, así pues, el desequilibrio de género en la muestra es atribuible a la elección diferencial de la informática que aún existe entre chicos y chicas. Adicionalmente, se aplica el instrumento en 2 colegios privados en los que pudimos comprobar 'in situ' su validez aparente (correcta visibilidad y navegabilidad del instrumento)

En la **Figura 4** se muestra la tasa de acierto por ítem, expresada en tanto por uno, confirmándose empíricamente la progresiva dificultad del instrumento; que fue ya anticipada por los expertos durante el proceso de validación de contenido (Román, 2015a). La tasa de acierto promedio a lo largo de los 28 ítems es de 0,59 (índice de dificultad medio); oscilando entre el valor 0,13 (índice de dificultad muy alto) del ítem 23 y el valor 0,96 (índice de dificultad muy bajo) del ítem 1.

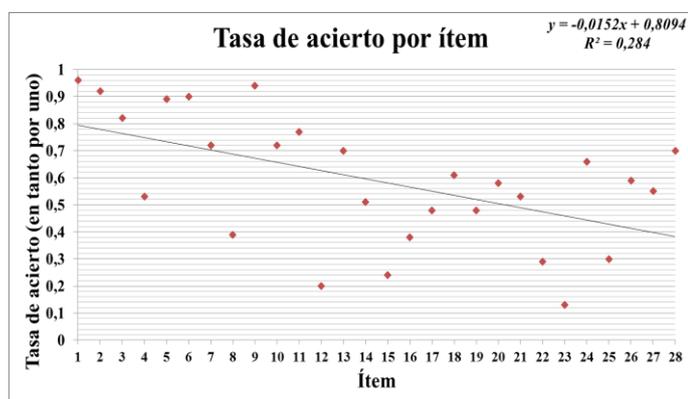


Figura 4. Tasa de acierto por ítem, expresada en tanto por uno.

La fiabilidad, como consistencia interna, del TPC en su conjunto arroja un valor de  $\alpha = 0,74$ ; que puede considerarse un valor aceptable. La puntuación total de cada sujeto en el TPC se calcula como suma de aciertos a lo largo de los 28 ítems del instrumento. En la **Tabla 2** se muestran los descriptivos relativos a las puntuaciones totales de nuestros sujetos en el TPC; y en la **Figura 5** se aporta un histograma con la distribución de dichas puntuaciones totales.

Tabla 2. Descriptivos relativos a la puntuación total en el TPC

N	Válidos	400
	Perdidos	0
Media		16,46
Error típico de la media		,213
Mediana		16,00
Moda		18
Desviación típica		4,266
Varianza		18,199
Asimetría		,050
Rango		21
Mínimo		6
Máximo		27
Percentiles	10	11,00
	20	13,00
	30	14,00
	40	15,00
	50	16,00
	60	17,00
	70	18,00
	80	20,00
	90	22,00

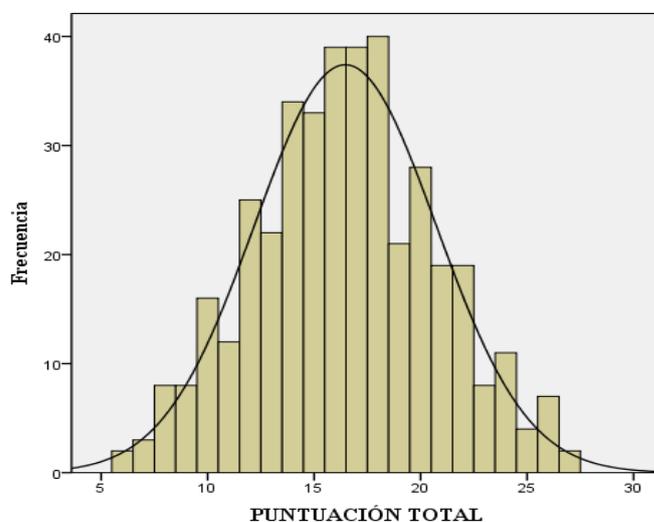


Figura 5. Histograma de puntuaciones totales en el TPC.

Para estudiar la normalidad de dicha distribución, se calculó la Z de Kolmogorov-Smirnov que arrojó un valor de  $Z_{k-s}=1,235$  con una probabilidad asociada de  $p=0,094 > 0,05$ . Podemos pues asegurar el ajuste de nuestra distribución a la curva normal.

En la **Tabla 3** se muestra la puntuación total promedio en el TPC en función del curso; hallándose diferencias muy significativas a favor de 2º ESO ( $t = 4,75$ ;  $p_{(0)} = ,000 < ,01$ ). En la **Tabla 4** se muestra la puntuación total promedio en el TPC en función del sexo; no hallándose diferencias significativas entre chicos y chicas ( $t = 0,67$ ;  $p_{(0)} = ,502 > ,01$ )

Tabla 3. Puntuación total en el TPC según curso.

	Curso	N	Media	Desv. típica	Error típ. de la media
Puntuación TPC	1ºESO	275	15,80	4,017	,242
	2ºESO	125	17,93	4,445	,398

Tabla 4. Puntuación total en el TPC según sexo.

	Sexo	N	Media	Desv. típica	Error típ. de la media
Puntuación TPC	Chico	261	16,57	4,318	,267
	Chica	139	16,27	4,175	,354

Adicionalmente, al final del TPC se les planteó a los sujetos un par de cuestiones sobre percepción de autoeficacia en su rendimiento en el test (“De 0 a 10, ¿cómo consideras que te ha salido el Test?”); y en su nivel de competencia informática (“De 0 a 10, ¿cómo consideras que se te dan los ordenadores y la informática?”). En la **Tabla 5** se muestran las correlaciones ( $r$ ), para el total de la muestra, entre la puntuación total en el TPC y las dos preguntas de autoeficacia.

Tabla 5. Correlación entre TPC y cuestiones de autoeficacia.

		¿Cómo consideras que te ha salido el Test?	¿Cómo consideras que se te dan los ordenadores y la informática?
Puntuación TPC	$r$	,324**	,128*
	Sig.	,000	,010
	N	400	400

Por último, se encontró una diferencia significativa a favor de los chicos frente a las chicas en la cuestión sobre autoeficacia del rendimiento en el test ( $X_{\text{chicos}} = 7,46$ ;  $X_{\text{chicas}} = 6,99$ ;  $t = 3,07$ ;  $p_{(0)} = ,002 < ,01$ )

## 5. CONCLUSIONES

Así pues, en este artículo se ha presentado el proceso de diseño de nuestro ‘Test de Pensamiento Computacional’ (TPC), y su comportamiento psicométrico general a partir de una aplicación sobre una muestra de 400 sujetos de 1º y 2º ESO. Podemos extraer las siguientes conclusiones:

- El TPC presenta un grado de dificultad adecuado (medio) para la población objetivo, con una dificultad creciente a lo largo de sus ítems; tal y como se recomienda en el diseño de pruebas de aptitud.
- Las puntuaciones totales en el TPC se distribuyen normalmente, y de manera simétrica, presentando una buena variabilidad que permite la construcción de baremos adecuados para la población de referencia.
- La fiabilidad del TPC ( $\alpha = 0,74$ ) es suficiente para ser una prueba de aptitud. En muestras adicionales que recién han realizado el TPC a través de tabletas (en vez de a través de ordenador fijo como hizo la muestra principal) hemos encontrado valores preliminares de fiabilidad  $\alpha \approx 0,85$ , sugiriendo que la precisión de nuestro test aumenta cuando se aplica sobre dispositivos móviles (quizás porque permiten al sujeto girar la pantalla a un lado y a otro, reduciendo la excesiva carga ‘espacial’ de los ítems del test).
- El rendimiento en el TPC se incrementa con el curso académico, algo coherente con nuestra hipótesis de que el PC es una aptitud cognitiva de solución de problemas que, por tanto, está vinculada con el desarrollo madurativo-evolutivo de los sujetos.

- El rendimiento en el TPC no está afectado por el sexo del sujeto; sin embargo, las chicas muestran significativamente menor confianza que los chicos sobre su respuesta al test. Este débil percepción de autoeficacia en tareas relacionadas con el PC podría afectar negativamente a las chicas al afrontar el aprendizaje de la programación, y podría disuadirlas de elecciones académicas y profesionales vinculadas a la informática; tal y como ya se ha sugerido en estudios previos de género (Soe & Yakura, 2008).
- Se encuentra una correlación significativa, positiva y de intensidad moderada entre la puntuación total en el TPC y la percepción de autoeficacia sobre el rendimiento en el mismo ( $r = ,324$ ); es decir, aunque los sujetos no tenían experiencia previa en pruebas de medida del PC, sí perciben de manera moderada que responden al test siguiendo una cierta lógica. En contraste, hay una correlación baja, aunque estadísticamente significativa, con la percepción de autoeficacia sobre su nivel general de competencia informática, posiblemente porque relacionan dicha competencia con usos-consumos pasivos de la tecnología en vez de con tareas activas y creativas como las que involucran la programación y PC.

De manera complementaria, estamos sometiendo al TPC a un proceso completo para estudiar su fiabilidad y validez. Como fuentes de validez concurrente del TPC estamos utilizando dos tipos de medidas: a) medidas ya validadas relativas a variables supuestamente próximas al PC, tales como ‘razonamiento lógico’, ‘resolución de problemas’ o ‘aptitudes perceptuales-atencionales’; b) medidas alternativas de PC, tales como selecciones de ítems y tareas que vienen siendo utilizados en concursos internacionales de promoción del talento computacional. Como fuentes de validez predictiva, pretendemos correlacionar las puntuaciones en el TPC de nuestros sujetos con sus medidas de desempeño en el programa ‘K-8 Intro to Computer Science’ (Code.org, 2015), y con la calidad de sus proyectos de final de curso realizados con Scratch medidos objetivamente con la herramienta analítica en desarrollo por Moreno & Robles (2014), Dr. Scratch (<http://drscratch.programamos.es/>)

Algunos de los usos a los que podría destinarse el TPC una vez completada su validación podrían ser: medidas pre-test del nivel inicial en PC de los estudiantes; detección temprana de estudiantes con altas aptitudes para tareas de programación informática; evaluación de currículums que tengan como objetivo la enseñanza-aprendizaje del PC; orientación académica y profesional de estudiantes hacia disciplinas STEM. Finalmente, aplicando el ‘Computational Thinking Framework (CTF)’ (Gouws, Bradshaw & Wentworth, 2013a) a nuestro instrumento, se hacen evidentes las siguientes limitaciones de nuestro TPC: al estar compuesto en su totalidad por ítems de elección múltiple y cerrada, sólo estaría midiendo el PC en sus niveles más bajos de complejidad cognitiva (‘reconocer’ y ‘comprender’). Un instrumento destinado a medir el PC también en sus niveles más altos de complejidad (‘aplicar’ y ‘asimilar’) deberá incluir adicionalmente ítems que soliciten no sólo reconocer sino también evocar el algoritmo correcto (como de hecho sugirió uno de nuestros expertos); y también problemas complejos y abiertos cuya resolución exija al estudiante transferir de manera creativa los conceptos del PC a diversos ámbitos.

REFERENCIAS

- Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832-835.
- Belshaw, D. (2013). *This is Why Kids Need to Learn to Code* [Mensaje en un blog]. Recuperado de <http://dmlcentral.net/blog/doug-belshaw/why-kids-need-learn-code>
- Brown, N. C. C., Kölling, M., Crick, T., Peyton Jones, S., Humphreys, S., & Sentance, S. (2013). Bringing computer science back into schools: Lessons from the UK. *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 269-274.
- Code.org (2015). *K-8 Intro to Computer Science* [Curso en línea]. Recuperado de <http://studio.code.org/s/20-hour>
- CSTA (2011). *K-12 Computer Science Standards (Level 2)* [Documento en línea]. Recuperado de [http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA\\_K-12\\_CSS.pdf](http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf)
- CSTA & ISTE (2011). *Operational Definition of Computational Thinking for K-12 Education* [Documento en línea]. Recuperado de <http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>
- European Schoolnet (2014). *Computing our future. Computer programming and coding: priorities, school curricula and initiatives across Europe* [Technical report]. Recuperado de <http://www.eun.org/publications/detail?publicationID=481>
- Gardner, H. (2014). *La generación APP: Cómo los jóvenes gestionan su identidad, su privacidad y su imaginación en el mundo digital*. Barcelona: Paidós.
- Google for Education (2015). *Exploring Computational Thinking* [Página web]. Recuperado de <https://www.google.com/edu/resources/programs/exploring-computational-thinking/>
- Gouws, L. A., Bradshaw, K., & Wentworth, P. (2013a). Computational thinking in educational activities: An evaluation of the educational game light-bot. *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, 10-15.
- Gouws, L., Bradshaw, K., & Wentworth, P. (2013b). First year student performance in a test for computational thinking. *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, 271-277.
- Grover, S., & Pea, R. (2013). Computational thinking in K-12. A review of the state of the field. *Educational Researcher*, 42(1), 38-43.
- Henderson, P. B., Cortina, T. J., & Wing, J. M. (2007). Computational thinking. *ACM SIGCSE Bulletin*, 39(1), 195-196.
- Howland, K., & Good, J. (2015). Learning to communicate computationally with flip: A bi-modal programming language for game creation. *Computers & Education*, 80, 224-240. doi: <http://dx.doi.org/10.1016/j.compedu.2014.08.014>
- Lee, G., Lin, Y., & Lin, J. (2014). Assessment of computational thinking skill among high school and vocational school students in Taiwan. *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 2014(1), 173-180.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61. doi: <http://dx.doi.org/10.1016/j.chb.2014.09.012>
- Manovich, L. (2013). *Software takes command*. New York: Bloomsbury.
- Moreno, J., & Robles, G. (2014). Automatic detection of bad programming habits in scratch: A preliminary study. *Frontiers in Education Conference (FIE), 2014 IEEE*, 1-4. doi:10.1109/FIE.2014.7044055
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York; Basic Books.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., et al. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60-67.
- Román, M. (2014). Aprender a programar 'apps' como enriquecimiento curricular en alumnado de alta capacidad. *Bordón. Revista de Pedagogía*, 66(4), 135-155.
- Román, M. (2015a). Computational Thinking Test: Design Guidelines and Content Validation. *7th annual International Conference on Education and New Learning Technologies*. IATED, Barcelona.
- Román, M. (2015b). *Test de Pensamiento Computacional* [Test en línea]. Recuperado de <http://goo.gl/IYEKMB>
- Rushkoff, D. (2010). *Program or be programmed*. New York: OR Books.
- Soe, L., & Yakura, E. K. (2008). What's wrong with the pipeline? Assumptions about gender and culture in IT work. *Women's Studies*, 37(3), 176-201.
- Walden, J., Doyle, M., Garns, R., & Hart, Z. (2013). An informatics perspective on computational thinking. *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, 4-9.
- Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The fairy performance assessment: Measuring computational thinking in middle school. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 215-220.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, 366(1881), 3717-3725. doi: <http://dx.doi.org/10.1098/rsta.2008.0118>