

**LA PROGRAMACIÓN, ¿UNA HERRAMIENTA
PARA DESARROLLAR LA CAPACIDAD
DE RESOLUCIÓN DE PROBLEMAS?**

**COMPUTER PROGRAMMING AS A TOOL
FOR DEVELOPING PROBLEM-SOLVING SKILLS**

Gara Miranda Valladares

gmiranda@ull.edu.es

Coromoto León Hernández

cleon@ull.edu.es

Casiano Rodríguez León

crguezl@ull.edu.es

Departamento de Ingeniería Informática y de Sistemas

Universidad de La Laguna

RESUMEN

El término pensamiento computacional se utiliza para describir cómo piensa un «científico/a de computadoras» cuando se enfrenta a un problema y los beneficios que esta forma de pensar podría tener en todos nosotros. El pensamiento computacional representa el proceso de reconocimiento de aspectos de la Informática en el mundo que nos rodea, y la aplicación de sus herramientas y técnicas para comprender y razonar sobre los sistemas y los problemas a los que nos enfrentamos.

El objetivo de este trabajo es analizar en qué medida influye el desarrollo del pensamiento computacional en la posterior mejora de habilidades relacionadas con la resolución de problemas en un ámbito científico, específico, o incluso general o de aplicación práctica en la vida cotidiana. Para ello, se ha trabajado con alumnado del Grado en Matemáticas y del Grado en Ingeniería Informática, titulaciones que, en menor y en mayor medida, incorporan asignaturas que introducen al alumnado en la «programación» (herramienta fundamental para todo «científico/a de computadoras»). La batería de tests EFAL (Evaluación Factorial de las Aptitudes Intelectuales) de nivel 4 se utilizó para medir la inteligencia general, la inteligencia no verbal (INV) y la inteligencia verbal (IV).

De los resultados obtenidos se puede deducir que existe una mayor homogeneidad en el desarrollo de la INV para el alumnado que presumiblemente ha recibido una mayor formación ligada a la programación. Aún así, y para poder obtener conclusiones más robustas será necesario extender el estudio a alumnado de otras ramas de conocimiento, comparando además las habilidades del alumnado de nuevo ingreso con la del alumnado que finaliza la titulación.

PALABRAS CLAVE: pensamiento computacional; habilidades cognitivas; desarrollo de software; informática.

ABSTRACT

The term computational thinking is used to describe how a «computer scientist» thinks when facing a problem and the benefits that this way of thinking could have in all of us. Computational thinking represents the process of recognizing aspects of computing in the world around us, and applying its tools and techniques to understand and think about the systems and problems we face.

The objective of this work is to analyze the extent to which the development of computational thinking influences the subsequent improvement of skills or competences related to the resolution of problems in a specific scientific field, or even in general and practical situations of everyday life. To this end, we have worked with students from the Degree in Mathematics and with students from the Degree in Computer Science. It's important to note that in both cases, these Degrees incorporate courses for introducing students to «computer programming» (a fundamental tool for all «computer scientist»). The EFAL (Factorial Assessment of Intellectual Abilities) test battery - level 4 - was used to measure general intelligence, non verbal intelligence (INV) and verbal intelligence (IV).

From the results obtained, it can be deduced that there is a greater homogeneity in the development of the INV for students who presumably have received more training linked to programming. Even so, and in order to obtain more robust conclusions, it will be necessary to extend the study to students from other branches of knowledge, comparing the skills of the younger students with the students who are close to finish the degree.

KEYWORDS: computational thinking; cognitive skills; software development; computer science.

INTRODUCCIÓN Y OBJETIVOS

Es sorprendente que siendo la Informática (Computer Science) 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.

Para trabajar en esa línea se está impulsando activamente un nuevo enfoque de enseñanza que incluya el pensamiento computacional (del inglés, computational thinking) en todos los niveles educativos. El pensamiento computacional podría describirse como los procesos de pensamiento implicados en la formulación de problemas y representación de sus soluciones, de manera que dichas soluciones puedan ser ejecutadas por un agente de procesamiento de información (ya sea un humano, un ordenador o combinaciones de ambos). Este término se hizo famoso gracias a un artículo (Wing, 2006) en el que se introduce el pensamiento computacional como un procedimiento que permite la resolución de problemas, el diseño de sistemas y la comprensión de la conducta humana haciendo uso de conceptos fundamentales de la Informática. Este proceso implica, por tanto, aprender a pensar sobre cómo representar y resolver problemas que requieren una combinación de potencia cognitiva humana y capacidad de cómputo. Este tipo de pensamiento es el que desarrollan de forma implícita quienes se dedican a la programación o desarrollo de aplicaciones informáticas. Quien es capaz de «programar» una computadora para que resuelva automáticamente un determinado problema, ha sido capaz de entender y modelar el problema en su sentido más amplio, proporcionando además un «algoritmo» o secuencia de instrucciones que lleven a su solución (Futschek, 2006). En este proceso interviene mucho la creatividad de la persona pero también la capacidad que ésta tenga para entender el problema, el entorno y ofrecer una solución al mismo. En este sentido, desde la investigación básica del razonamiento humano, se ha

puesto de relieve que el cerebro es un mecanismo de cómputo bien basado en modelos mentales (Johnon-Laird & Khemlani, 2013), en reglas (Rips, 2013), o en algoritmos conformados por la evolución para la resolución de problemas adaptativamente relevantes (Cosmides & Tooby, 2013).

Desde 2006, momento en el que se utiliza el término *computational thinking* por primera vez, el pensamiento computacional ha atraído su atención en el contexto de la educación primaria y secundaria, principalmente en países anglosajones, como Estados Unidos. No obstante, aún no existe consenso alguno sobre la definición del concepto de pensamiento computacional, habiendo múltiples variantes (Barr & Stephenson, 2011; Brennan & Resnick, 2012; Grover & Pea, 2013). Por ejemplo, la ISTE (International Society for Technology in Education) junto con la CSTA (Computer Science Teachers Association), definen el pensamiento computacional como un proceso de solución de problemas que incluye, pero no se limita, a las siguientes dimensiones:

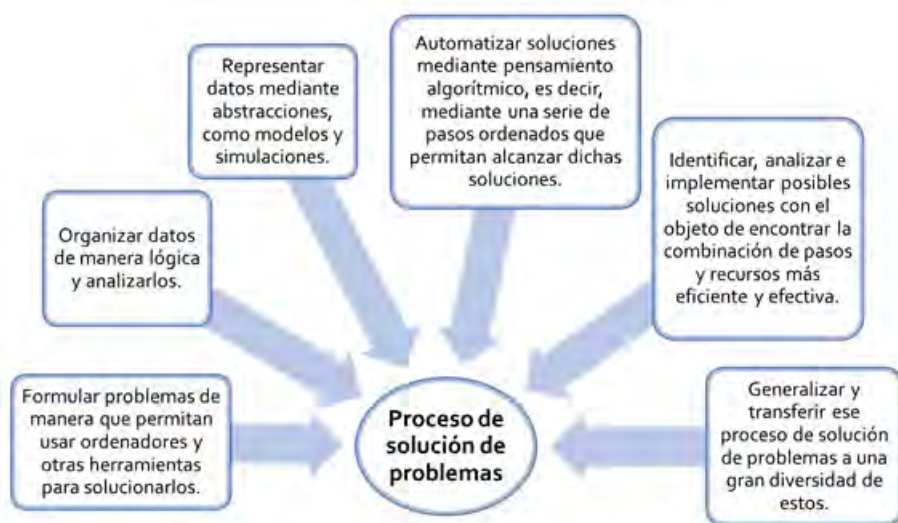


Figura 1. El pensamiento computacional como proceso de resolución de problemas

Quienes fomentan e impulsan la incorporación del pensamiento computacional en los sistemas educativos actuales, consideran que no es simplemente un proceso de solución de problemas sino que es una habilidad intelectual básica, comparable a la lectura, la escritura, la expresión oral o la aritmética. Estas habilidades básicas son fundamentales e imprescindibles para describir y explicar a los demás problemas o situaciones complejas y el pensamiento computacional sirve exactamente para eso mismo. Podríamos decir entonces que el pensamiento computacional es comparable a las otras habilidades cognitivas básicas que cualquier

persona, en nuestra sociedad moderna, debería poseer. Al fin y al cabo, el pensamiento computacional es otro lenguaje (además del escrito y hablado, así como el matemático) que los humanos podemos utilizar para hablar del universo y de sus procesos.

En este sentido, desde 2012, el Consejo Nacional de Investigación (del inglés, National Research Council, NRC) de Estados Unidos recomienda formalmente las matemáticas y el pensamiento computacional como unas de las ocho principales prácticas en el ámbito de las ciencias y la ingeniería (National Research Council, 2012):

- 
1. Asking questions and defining problems
 2. Developing and using models
 3. Planning and carrying out investigations
 4. Analyzing and interpreting data
 - 5. Using mathematics and computational thinking**
 6. Constructing explanations and designing solutions
 7. Engaging in argument from evidence
 8. Obtaining, evaluating, and communicating information

Figura 2. Prácticas recomendadas por el NRC en el ámbito de las ciencias y la ingeniería

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 ello que consideramos primordial llevar a cabo una campaña de difusión y de sensibilización en el sector educativo. Creemos que, como primer paso, es necesario dar visibilidad al papel de la Informática, en el mundo que nos rodea, luchando contra algunos estereotipos que se han forjado en este sector. Es importante que los jóvenes comprendan la repercusión que los avances en este campo han tenido en la calidad de vida actual y en el estilo de vida moderno. La Informática en particular, y las Nuevas Tecnologías en general, han tenido un alto impacto en el desarrollo y en el avance de múltiples y muy diversos campos de estudio. Es por tanto, una herramienta de importante conocimiento no sólo para quienes vayan a dedicarse a este ámbito en concreto sino para quienes quieran avanzar e innovar en cualquier otro ámbito de conocimiento.

Ahora bien: ¿cómo implementar y mejorar el pensamiento computacional de nuestros escolares? La programación es una tarea fundamental en el ámbito de la Informática, pues nos permite implementar y llevar a la realidad aquellas herramientas o aplicaciones informáticas que se hayan diseñado o modelado previamente. La programación supone en sí misma una forma de pensamiento computacional, pues representa el mecanismo mediante el cual los expertos informáticos materializan el proceso de analizar un problema, modelarlo y expresar una solución del mismo de tal forma que pueda ser resuelto automáticamente por un sistema computacional. Es por ello que la programación es el eje principal sobre el cual se desarrolla y fomenta el pensamiento computacional entre los más jóvenes (Lye & Koh, 2014).

Algunas de las definiciones de pensamiento computacional consideran que los estudiantes hacen uso del pensamiento computacional incluso cuando no están utilizando algún tipo de herramienta informática. Contrariamente, la programación sí que implica que los estudiantes hagan uso del pensamiento computacional a través de la construcción de artefactos (Resnick et al., 2009; Kafai & Burke, 2013).

La mayoría de las propuestas para fomentar el pensamiento computacional se basan en la utilización de herramientas o lenguajes de programación de tipo visual (López, 2014). En este sentido, una de las herramientas más extendidas es Scratch. Scratch comparte características de los lenguajes de programación visual modernos, los cuales son fáciles de aprender, aparte de que proporcionan retroalimentación visual de los programas desarrollados en forma de objetos animados, permitiendo al alumnado la creación de medios interactivos, como por ejemplo, animaciones y juegos. Otro añadido es que Scratch es probablemente el lenguaje de programación visual más apropiado para desarrollar el pensamiento computacional a través de la programación en el ámbito de la educación primaria y secundaria (Baytak & Land, 2011; Brennan & Resnick, 2012; Kafai, Fields, & Burke, 2011; Tangney et al., 2010; Theodorou & Kordaki, 2010). Tal y como Mitch Resnic (Director del Grupo Lifelong Kindergarten del MIT Media Lab) menciona en su charla TED «Aprender a Programar, Programar para Aprender», la programación, es decir, la habilidad para crear programas informáticos es una parte fundamental del

alfabetismo en la sociedad de hoy en día: «cuando los estudiantes aprenden a programar en Scratch, aprenden importantes estrategias para la resolución de problemas, para el diseño de proyectos y para la comunicación de ideas».

El objetivo de este proyecto consistía en analizar en qué medida influye el desarrollo del pensamiento computacional en la posterior mejora de habilidades relacionadas con la resolución de problemas en un ámbito científico, específico, o incluso general o de aplicación práctica en la vida cotidiana. Para ello, durante el proyecto se trabajó con alumnado del Grado en Matemáticas y del Grado en Ingeniería Informática, titulaciones que, en menor y en mayor medida, incorporan asignaturas que introducen al alumnado en la «programación» (herramienta fundamental para desarrollar el pensamiento computacional).

METODOLOGÍA

Para analizar, de forma general, la capacidad de resolución de problemas, se decidió hacer uso de tests comerciales, ampliamente utilizados en distintos ámbitos, y correctamente validados. De los tests disponibles en el mercado, y teniendo en cuenta el tipo de alumnado al que irían dirigido y el tipo de habilidades a analizar, se decidió utilizar la batería de tests EFAL (Evaluación Factorial de las Aptitudes Intelectuales) de nivel 4. EFAL es una batería de aplicación de referencia en la evaluación de las aptitudes intelectuales que permite evaluar de una forma muy completa y con un enfoque homogéneo a personas con diferente formación. Evalúa la capacidad para resolver ágilmente problemas de muy diverso tipo, mantener una adecuada flexibilidad intelectual y realizar procesos lógicos de deducción e inducción. Está formado por cuatro baterías, cada una dirigida a un nivel de edad o de formación diferente, y compuestas por cinco tests: aptitud espacial, aptitud numérica, razonamiento abstracto, razonamiento verbal y memoria.

Así mismo ofrece puntuaciones en inteligencia general, inteligencia no verbal e inteligencia verbal, así como una serie de índices del estilo de respuesta del sujeto (rapidez/eficacia). La inteligencia general se presenta como cociente intelectual, la inteligencia no verbal mide la capacidad para razonar y resolver problemas de tipo abstracto o figurativo; la capacidad para analizar y transformar mentalmente estímulos visuales y gráficos; la agilidad para resolver problemas de tipo lógico y abstracto; el pensamiento computacional y la habilidad para analizar, sintetizar y comparar información de tipo espacial. La inteligencia verbal mide la capacidad para expresar, comprender y analizar información verbal y numérica; capacidad para razonar con conceptos verbales o numéricos; capacidad para captar relaciones entre distintos conceptos y trabajar con ellos; habilidad para interpretar y extraer información de gráficos y tablas con información numérica; agilidad de cálculo y amplio vocabulario. En estos casos, las posibles categorías para los resultados son: baja, media-baja, media, media-alta y alta. En cuanto al estilo de respuesta, la rapidez evalúa el número de elementos que el

sujeto intenta y está relacionada con la velocidad con la que procesa los distintos problemas, mientras que la eficacia es el porcentaje de elementos intentados que acierta el sujeto.

Las pruebas se recogen en un único cuadernillo y se evalúan con ítems variados, de gran capacidad de discriminación y presentados a todo color. El tiempo de aplicación es breve aunque, para garantizar la calidad de la evaluación, cada subtest se aplica con su propio tiempo. Los amplios estudios de tipificación (N= 23.793), fiabilidad y validez que lo acompañan, junto con la completa norma de aplicación e interpretación, convierten al EFAI en un interesante y relevante instrumento de la evaluación de las aptitudes intelectuales.

En la evaluación participó un grupo de 67 alumnos/as de la asignatura «Técnicas Experimentales» (segundo cuatrimestre del primer curso del Grado en Matemáticas) y un grupo de 55 alumnos/as de la asignatura «Lenguajes y Paradigmas de Programación» (primer cuatrimestre del tercer curso del Grado en Ingeniería en Informática). En el grupo de primero de Matemáticas, un 55.2% de los participantes eran hombres y un 44.8% mujeres, mientras que en el grupo de tercero de Ingeniería Informática el 89.1% de los participantes eran hombres frente al 10.9% de mujeres.

Cabe destacar que cada participante realizó de forma individual el test EFAI de nivel 4 y que en ningún caso los participantes tuvieron información previa del test, para evitar posibles preparaciones específicas para el test, que pudieran interferir en los resultados.

TABLA 1: DESCRIPCIÓN DE LA MUESTRA PARA LA REALIZACIÓN DE LA PRUEBA

TITULACIÓN	GRADO EN ING. INFORMÁTICA	GRADO EN MATEMÁTICAS
Asignatura	Lenguajes y Paradigmas de Programación	Técnicas Experimentales
Curso	Tercero	Primero
Cuatrimestre	Primero	Segundo
N.º alumnos matriculados	160	90
N.º alumnos que realizaron el test	55	67
Datos de género en % (H/M)	89,1 / 10,9	55,2 / 44,8

RESULTADOS

La evaluación del test EFAI-4 da como resultado una evaluación final del individuo en la que se obtienen puntuaciones diferenciadas para los aspectos siguientes: aptitud espacial (Esp.), aptitud numérica (Num.), razonamiento abstracto (Raz.), aptitud verbal (Ver.), memoria (Mem.), inteligencia no verbal (INV), inteligencia verbal (IV), cociente intelectual (CI), rapidez (RAP) y

eficacia (EFI). Para los primeros 5 aspectos se proporcionan las puntuaciones directas, es decir, el número de aciertos obtenidos por el sujeto en cada caso. El número total de ítems para cada uno de los aspectos valorados es: Esp. = 22, Num. = 25, Raz. = 25, Ver. = 22, Mem. = 20.

Si nos fijamos en la Tabla 2, comprobaremos que en general, los valores medios para estos aspectos están muy por debajo del número de ítems en cada caso. Los peores resultados se obtienen en la aptitud verbal mientras que los mejores se obtienen en aptitud numérica. Esto encaja con el hecho de que los sujetos proceden de áreas de conocimiento de ciencias e ingeniería y no de humanidades ni ciencias sociales.

TABLA 2: RESULTADOS OBTENIDOS

	ESP.	NUM.	RAZ.	VER.	MEM.	INV	IV	CI	RAP	EFI
GRADO DE INGENIERÍA INFORMÁTICA										
Media	7,5	10,1	7,9	6,6	10,3	9,1	7,4	90,3	73,8	44,3
Min.	1,0	5,0	2,0	2,0	3,0	3,0	3,0	71,0	32,0	23,5
Máx.	13,0	18,0	15,0	14,0	18,0	15,0	14,0	122,0	99,0	76,0
Desv.	2,9	3,0	2,6	2,7	3,5	2,7	2,6	11,6	13,0	9,8
GRADO EN MATEMÁTICAS										
Media	7,3	10,1	8,2	6,8	11,4	9,2	7,4	90,5	59,8	55,6
Min.	2,0	3,0	1,0	2,0	2,0	3,0	2,0	64,0	27,0	27,0
Máx.	16,0	22,0	22,0	13,0	19,0	18,0	16,0	135,0	114,0	85,0
Desv.	3,1	4,0	3,5	2,8	3,6	3,5	3,2	15,0	17,0	14,1

Para la INV los resultados obtenidos en la muestra de alumnado de Matemáticas corresponden a un valor Medio (percentiles 30-70 en una distribución normal) con un 51% de los participantes en esta categoría. En el caso de Ingeniería Informática también corresponde al nivel Medio con un 56% de los participantes en esta categoría. Para la IV los resultados corresponden a la categoría Baja (percentiles 5-17 en una distribución normal) con un 43% de los participantes en esta categoría para el caso de Matemáticas y de un 45% para el caso de Ingeniería Informática. En lo referente a la comparación entre ambos tipos de inteligencias, se observa que en el 60% de los participantes tiene una puntuación en INV significativamente mayor que en IV.

El estilo de respuesta más predominante en la muestra analizada es el de Pausado e Ineficaz (con un 46% de los participantes) para el caso de Matemáticas y de Rápido e Ineficaz (con un 58% de los participantes) para el caso de Ingeniería Informática. Por último, el rango de CI más predominante en la muestra analizada es el correspondiente a la Media (91-110) con el

35% de los participantes para el caso de Matemáticas y con el 38% de los participantes para el caso de Ingeniería Informática.

Si comparamos los valores promedios de las dos muestras, para cada una de las capacidades medida, podemos ver cómo los valores son muy similares en casi todos los casos. La mayor diferencia se encuentra en el resultado para «Memoria» y para tipo de respuesta «Rapidez/Eficacia». Los estudiantes del Grado en Matemática obtienen una mayor puntuación media en memoria, una menor puntuación en rapidez y una mayor puntuación en eficacia. Centrándonos en el indicador de inteligencia no verbal (INV), que mide principalmente la capacidad para razonar y resolver problemas de tipo abstracto o figurativo, los resultados obtenidos para el alumnado de Ingeniería Informática y de Matemáticas están ambos en la categoría «media». En ambos casos, el valor promedio para la INV es muy similar (9,1 - 9,2) para las dos muestras pero la desviación estándar es mayor para el alumnado de Matemáticas (3,5 frente a 2,7). Si nos fijamos en el resto de indicadores, podremos observar que la desviación estándar siempre mayor para la muestra de Matemáticas.

Para analizar un poco mejor esta desviación en los resultados, en la Figura 3 se muestra, para ambas titulaciones, el número medio de estudiantes (eje y) que obtiene cada una de las posibles puntuaciones de INV (eje x). En general, podemos comprobar que los resultados son más uniformes para el caso de Ingeniería Informática: existen menos casos en los extremos y muchos más casos en los valores intermedios. Sin embargo, los resultados para el caso de Matemáticas reflejan una distribución mucho menos uniforme, con más casos en los extremos y con una menor homogeneidad en los valores intermedios.

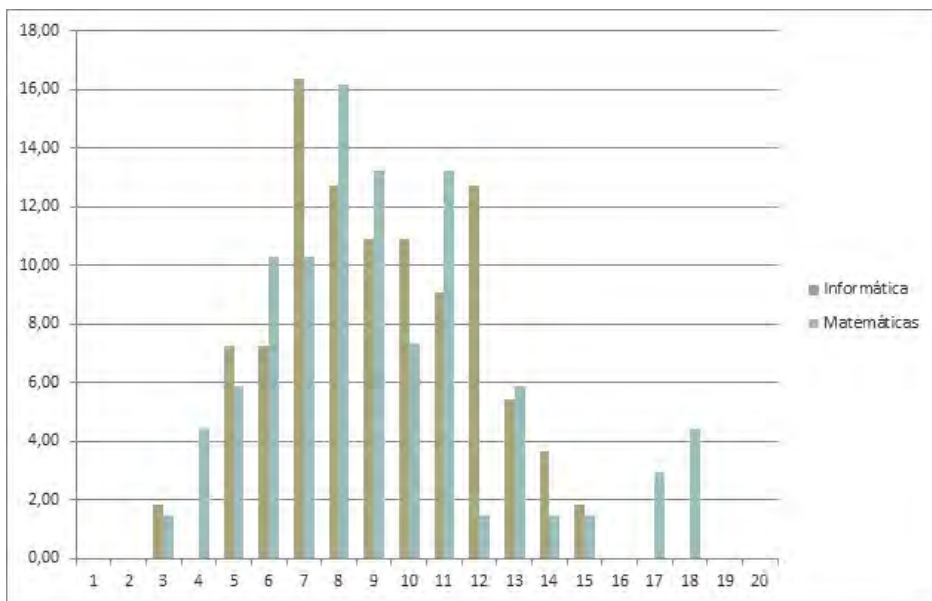


Figura 3: Número de individuos para cada una de las puntuaciones obtenidas en INV

CONCLUSIONES

El objetivo de este proyecto de investigación no era aplicar nuevas estrategias o metodologías que tuvieran un impacto en la adquisición de competencias en el alumnado, sino tratar de averiguar si la formación que se viene llevando a cabo en titulaciones diferentes - pero de ramas de conocimiento afines - tiene un efecto significativo en el desarrollo de algunas habilidades o competencias básicas como son la capacidad para solucionar problemas, el razonamiento, la creatividad, etc.

De los resultados obtenidos tras la ejecución de este proyecto, podríamos deducir que existe una mayor homogeneidad en el desarrollo de la INV para el alumnado que presumiblemente ha recibido una mayor formación ligada a la programación. El alumnado de Matemáticas estaba terminando su primer año de Grado mientras que el de Ingeniería Informática estaba comenzando su tercer año de Grado. Esta diferencia pudiera ser la responsable de la menor homogeneidad entre el alumnado de Matemáticas. Sin embargo, no sabemos si realmente existe una correlación entre la formación recibida por el alumnado de Ingeniería Informática y la homogeneidad observada en las habilidades analizadas. Pudiera ser que esta homogeneidad ya existiera cuando el alumnado inició estos estudios.

Es por este motivo que, como trabajo futuro, consideramos indispensable extender este análisis entre alumnado de más titulaciones (incluyendo Grados de otras ramas de conocimiento), analizando siempre dos grupos de alumnado: los que acaban de iniciar la titulación y los que están próximos a finalizarla. De esta forma, no sólo podremos estudiar las diferencias entre las distintas titulaciones/ramas de conocimiento, sino también los efectos de cada titulación en las distintas habilidades intelectuales analizadas.

REFERENCIA BIBLIOGRÁFICAS

- BARR, V. & STEPHENSON, C. (2011). *Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community?* ACM Inroads, 2(1), 48-54.
- BAYTAK, A. & LAND, S. (2011). *An investigation of the artifacts and process of constructing computers games about environmental science in a fifth grade classroom.* Educational Technology Research and Development, 59(6), 765-782.
- BRENNAN, K. & RESNICK, M. (2012). *Proceedings of the Annual American Educational Research Association Meeting: New frameworks for studying and assessing the development of computational thinking.* Vancouver, Canada.
- COSMIDES, L. & TOOBY, J. (2013). *Evolutionary psychology: New perspectives on cognition and motivation.* Annual Review of Psychology, 64, 201-229.

- FUTSCHEK, G. (2006). *Algorithmic thinking: The key for understanding computer science*. Informatics Education - The Bridge between Using and Understanding Computers, LNCS 4226, 159-168.
- GROVER, S. & PEA, R. (2013). *Computational Thinking in K-12: A Review of the State of the Field*. Educational Researcher, 42(1), 38-43.
- JOHNSON-LAIRD, P. N. & KHEMLANI, S. S. (2013). *Toward a unified theory of reasoning*. En B. Ross (Eds), *The Psychology of Learning and Motivation* (pp. 1-42). Elsevier.
- KAFAI, Y. B. & BURKE, Q. (2013). *Computer Programming Goes Back to School*. Phi Delta Kappan, 95(1), 61-65.
- KAFAI, Y. B., FIELDS, D. A. & BURKE, W. Q. (2011). *Entering the clubhouse: case studies of young programmers joining the online scratch communities*. Journal of Organizational and End User Computing, 22(2), 21-35.
- LÓPEZ, J. C. (2014). *Impacto de Scratch en el desarrollo del pensamiento algorítmico*. Master Thesis. Universidad ICESI.
- LYE, S. Y. & KOH, J. H. (2014). *Review on teaching and learning of computational thinking through programming: What is next for K-12?* Computers in Human Behavior, 41(0), 51-61.
- NATIONAL RESEARCH COUNCIL. (2012). *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. Washington, D. C., USA. The National Academies Press.
- 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.
- RIPS, L. J. (2013). *How many is a zillion? sources of number distortion*. Journal of Experimental Psychology: Learning, Memory, and Cognition, 39, 1257-1264.
- TANGNEY, B., OLDHAM, E., CONNEELY, C., BARRETT, S. & LAWLOR, J. (2010). *Pedagogy and Processes for a Computer Programming Outreach Workshop: The Bridge to College Model*. IEEE Transactions on Education, 53(1), 53-60.
- THEODOROU, C. & KORDAKI, M. (2010). *Super Mario: a collaborative game for the learning of variables in programming*. International Journal of Academic Research, 2(4), 111-118.
- WING, J. (2006). *Computational Thinking*. Communications of the ACM, 3(39): 33-35.