

# Validación de la escala de actitud ante la programación

Montse Garcia-Famoso<sup>1</sup>

Jorge-Manuel Dueñas<sup>2</sup>

M.Àngels Moncusí<sup>1</sup>

<sup>1</sup>Departament d'Enginyeria Informàtica i Matemàtiques;

<sup>2</sup>Departament de Psicologia

Universitat Rovira i Virgili

43007 Tarragona

montse.garcia@urv.cat

jorgemanuel.duenas@urv.cat

angels.moncusi@urv.cat

## Resumen

Existen muchos estudios que evalúan la dificultad que suponen los cursos de programación para los alumnos, en todos los niveles. Nosotros hemos observado que, en nuestra Universidad, el fracaso en la asignatura de Fundamentos de programación suele ser mayor en aquellos alumnos de las ingenierías menos relacionadas con la informática o las telecomunicaciones. Diversos estudios establecen una correlación entre el rendimiento de los alumnos y su percepción personal de competencia en programación. Para poder actuar de manera adecuada, y tratar de prevenir el fracaso, es imprescindible poder hacer un buen diagnóstico previo a los alumnos. Existen diversas encuestas para valorar la actitud de los alumnos ante la programación, sin embargo, la mayoría de estas encuestas, o no se encuentran disponibles o se trata de encuestas en inglés. En cualquier caso, antes de poder utilizar cualquiera de ellas, es necesario validarlas para nuestro contexto cultural. En este trabajo presentamos la adaptación y validación del instrumento *Computer Programming Attitude Scale* (CPAS) para estudiantes universitarios. Tras realizar el análisis factorial exploratorio podemos concluir que presenta unas propiedades psicométricas adecuadas para ser utilizado en la población universitaria española.

## Abstract

There are many studies that evaluate the difficulty of computer programming courses for students, at all levels. We have observed that, in our university, dropout in the introductory computer programming subject tends to be greater in those students less related to computing or telecommunications engineering. Several studies establish a correlation between students learning performance and their personal perception of competence in programming. In order to be able to act appropriately and try to prevent failure, it is essential to make a good prior diagnosis of the students. There

are several surveys to assess the attitude of students to programming, however, most of these surveys are either not available or are surveys in English. In any case, before being able to use any of them, it is necessary to validate them for our cultural context. In this paper we present the adaptation and validation of the *Computer Programming Attitude Scale* (CPAS) instrument for university students. After performing the exploratory factor analysis, we can conclude that it presents adequate psychometric properties to be used in the Spanish university population.

## Palabras clave

Escala de actitud, CPAS, programación, educación universitaria, validez, fiabilidad, auto-percepción, motivación.

## 1. Introducción

Los cursos de programación suelen resultar difíciles para los estudiantes debido a que requieren un alto nivel de abstracción así como de capacidad en la resolución de problemas [18]. Son difíciles, sobre todo, para alumnos de primer curso, lo que en ocasiones provoca un alto grado de suspensos y de abandonos. En [11] se presenta una revisión de las distintas problemáticas con las que se encuentran los estudiantes en los cursos de introducción a la programación, así como de las distintas estrategias pedagógicas que se han intentado para solucionarlas.

Algunos estudios se han centrado en intentar determinar aquellos conceptos de programación que pueden resultar más difíciles de comprender por parte de los estudiantes [22, 30, 34]. Otros autores introducen metodologías activas y centradas en el estudiante, [1, 29] como por ejemplo, técnicas de programación ágil, el SCRUM, aprendizaje basado en proyectos o el uso de la clase invertida.

Otros autores [24] proponen usar soluciones tecnológicamente más atractivas como el Logo y el Scratch, que complementan con una plataforma propia, el "Algorithmi", que usan de apoyo al aprendizaje de la algorítmica minimizando así, los errores en los algoritmos y corrigiéndolos de manera automatizada.

En otra serie de estudios, observan que el problema está en la dificultad que tienen los estudiantes para obtener una retroalimentación rápida y efectiva, lo que conlleva a una pérdida en la motivación, [29] propone la evaluación por pares, sin embargo, [7] considera la dificultad de obtener una retroalimentación correcta en este tipo de evaluación y proponen usar en las correcciones "alumnos" simulados. De forma similar, [4] utiliza la plataforma *On-line Judge* para dar una evaluación inmediata de los ejercicios propuestos y así reforzar la confianza de los alumnos en sus habilidades como programadores.

Hay autores que observan que los estudiantes tienden a experimentar miedo y falta de motivación para aprender a programar, y proponen diversas soluciones como por ejemplo, un asistente inteligente que responde a las preguntas sobre programación de los alumnos de forma inmediata [19], o el uso de una plataforma de juego para que les permita aprender mientras se divierten [6]. Por otro lado, observan que hay una tasa de abandono en los cursos de programación muy elevada y proponen mejorar la actitud de los estudiantes hacia el aprendizaje promoviendo el uso de libros electrónicos de auto-aprendizaje personalizados según la experiencia del estudiante [16].

Desde nuestro punto de vista, resulta imprescindible realizar un análisis previo que pueda ofrecernos una visión particular de nuestros alumnos para poder prever las medidas a aplicar en el curso. Por lo tanto, resulta imprescindible disponer de herramientas validadas para realizar dicho análisis [21], tal y como presentamos en la sección 2. El resto del artículo se organiza de la siguiente manera: En la sección 3 realizamos una breve revisión de la utilización de encuestas en el contexto de la programación. En la sección 4 explicamos la metodología llevada a cabo para validar el cuestionario. En la sección 5 mostramos los resultados obtenidos y en la sección 6 presentamos las conclusiones. Finalmente, en la sección 7 se indican futuras líneas de trabajo. En el anexo A incluimos la versión final del instrumento.

## 2. Motivación

De igual manera que sucede en otras universidades, el curso de introducción a la programación presentan altos índices de fracaso. En nuestro caso concreto, se trata de una asignatura, de primer curso, común a múltiples grados de ingeniería: desde ingeniería eléctrica hasta ingeniería informática. Esto hace que haya una

gran diversidad en cuanto a la formación previa, intereses personales y motivación de los alumnos.

Gomes, Santos y Mendes [12], demuestran que existe una gran correlación entre el rendimiento de los alumnos y la percepción personal de su competencia en programación. En nuestro contexto particular, con una alta heterogeneidad de alumnos, la actitud ante la programación de los alumnos puede ser un elemento decisivo en los resultados de los mismos, por lo que resulta imprescindible realizar un buen diagnóstico previo sobre dicha percepción. Dicho diagnóstico puede ayudarnos a tomar medidas más apropiadas, precisas y concretas para reducir el fracaso.

La escala de actitud ante la programación (*Computer Programming Attitude Scale - CPAS*) [3] se ha demostrado como un instrumento válido y de confianza para evaluar la actitud de los alumnos ante la programación, sin embargo, dicha escala se ha desarrollado en un contexto cultural diferente del español, lo que supone que antes de poder utilizarla hay que adaptarla, y por lo tanto, resulta necesario volver a evaluar su validez y fiabilidad en el nuevo contexto [21]. En este artículo se presenta la adaptación de la CPAS y su validación para poder utilizarla dentro del contexto universitario español.

## 3. Revisión de la literatura

Hemos llevado a cabo una búsqueda de artículos relacionado con las escalas y cuestionarios utilizados en el campo de la programación. El objetivo principal de la búsqueda no ha sido realizar una revisión exhaustiva de la literatura, si no tener una visión inicial tanto de las escalas propuestas, como de su utilización en el campo de la programación.

La búsqueda se ha realizado únicamente en la base de datos Scopus, ya que incluye las principales bases de datos, entre otras, IEEE, ACM, Elsevier, Springer. Hemos utilizando cualquier combinación de los siguientes términos en el título, en el resumen o en las palabras clave: ("*programming course?*") OR ("*computer programming*") OR ("*computer educatio*") AND ("*self-efficacy*") OR ("*attitude*") AND ("*scale*") AND ("*learning*").

El resultado ha sido de un total de 38 artículos, de los cuales se han descartado 23 entradas, porque no se ajustaban al objetivo de la búsqueda o porque no hemos podido acceder a ellos ya que estaban fuera de nuestro alcance.

Algunos de los artículos revisados diseñan sus propias herramientas de evaluación a partir de encuestas *ad hoc* o entrevistas [8, 13, 32].

Otra parte de los artículos revisados utilizan diversos instrumentos validados para evaluar diversos aspectos relacionados con la programación:

- Evaluación de la actitud o de la percepción de la eficacia respecto a la programación, la programación de móviles o su aprendizaje [2, 5, 9, 10, 14, 17, 27, 35, 36], los ordenadores [26].
- Evaluación de la actitud hacia los entornos de aprendizaje personalizados [28].
- Evaluación de la percepción de los estudiantes respecto a los laboratorios de informática [26].
- Escalas para evaluar la ansiedad respecto a la informática [17, 33].
- Evaluación de los estilos de aprendizaje [27, 35].

Uno de los artículos [15] presenta la validación de la adaptación cultural a Turquía de un cuestionario previo. Adapta la escala de auto-eficacia hacia las tecnologías en línea (*Online Technologies Self-Efficacy Scale - OTSES*), desarrollada originalmente por M. Miltiadou (2003).

Finalmente, Ö. Korkmaz, R. Çakir y M.Y. Özden [20] desarrollan y validan una escala para evaluar el nivel de pensamiento computacional (*CTS - Computational thinking scales*). Se trata de una encuesta de 29 ítems organizados en 5 factores: creatividad, pensamiento algorítmico, cooperatividad, pensamiento crítico y resolución de problemas.

Después de hacer una primera revisión de la literatura, se ha podido constatar que, existen diversos cuestionarios validados en otros países, pero no se ha encontrado ninguno validado para utilizar en el contexto universitario español.

## 4. Método

### 4.1. Participantes

Dado que el objetivo principal de la investigación consiste en la adaptación del cuestionario *Computer Programming Attitude Scale* (CPAS) [3], se recogieron datos de los estudiantes de la Universidad Rovira i Virgili. La muestra consta de 370 estudiantes universitarios (73,02 % hombres, 26,89 % mujeres, 0,5 % prefirió no decirlo), de los cuales, el 85,14 % son alumnos de primero (Fundamentos de programación) y el 14,86 % son de segundo (Programación), de diversos grados de ingeniería según la distribución mostrada en la Figura 1. El rango de edad de los sujetos está comprendido entre los 17 y los 48 años, con una media de edad de 19,66 años (D.T. = 3,30).

### 4.2. Instrumentos

Se utilizó el *Computer Programing Attitude Scale* (CPAS) [3] para estudiantes universitarios que consta de 18 ítem en escala *Likert*, con cinco alternativas de respuesta: (1) Completamente en desacuerdo, (5)

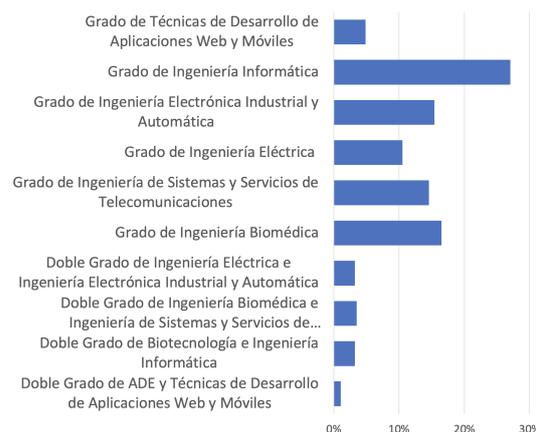


Figura 1: Distribución de estudiantes encuestados según el grado cursado.

Completamente de acuerdo. Seis de los ítems corresponden al factor afecto, seis al factor cognición y los seis restantes al factor comportamiento. El instrumento mide las actitudes hacia la programación en estudiantes universitarios.

Utilizamos un cuestionario *ad hoc* para evaluar la actitud hacia la programación, mediante tres ítems: 1) En general, tengo creencias positivas hacia la programación; 2) En general, tengo sensaciones agradables hacia la programación; 3) Habitualmente me siento cómodo/a realizando actividades relacionadas con la programación. Para responder se utilizó una escala *Likert* con cinco alternativas de respuesta: (1) Completamente en desacuerdo, (5) Completamente de acuerdo. Además, se preguntó el sexo, la edad, la educación previa al ingreso universitario, y el curso actual.

### 4.3. Procedimientos

Los cuestionarios se administraron colectivamente a través del espacio de las asignaturas en la plataforma de docencia virtual de la Universidad. Se garantizó el derecho al anonimato de los alumnos y la confidencialidad de sus resultados particulares. Se proporcionó el tiempo necesario para contestar los cuestionarios, sin límite de tiempo, con autorización voluntaria por parte de los estudiantes. Se les proporcionaron instrucciones sobre la manera adecuada de responder.

La traducción del instrumento original del inglés al castellano se realizó en cuatro fases. Primero, se realizó la traducción mediante el procedimiento de *back-translation* descrito por [25], un traductor bilingüe que domina el español y el inglés tradujo el CPAS al español. A continuación, un hablante nativo de inglés y traductor profesional, que no estaba familiarizado con la versión original del CPAS, volvió a traducir la versión en español al inglés, posteriormente otro traductor profesional evaluó si los ítems en español eran fieles al

significado de los ítems originales. Segundo, dos investigadores, un experto en evaluación psicológica y otro experto en programación revisaron la idoneidad, claridad, y expresiones culturales de los ítems. Tercero, se realizó una prueba de jueces. Específicamente, se seleccionó a seis investigadores externos, tres expertos en evaluación psicológica y tres en programación. Cuarto, se realizó una prueba piloto con un grupo de 35 estudiantes, a los cuales se les preguntó sobre la claridad de los ítems.

#### 4.4. Análisis de datos

Respecto al análisis de los datos, se llevó a cabo un análisis factorial exploratorio (AFE) para determinar el número más apropiado de factores subyacentes a los datos e identificar cualquier ítem que funcione mal y que haya que eliminar, debido a que nuestros datos mostraron una estructura unifactorial, no fue necesario realizar un análisis factorial confirmatorio. El AFE se realizó utilizando FACTOR 10.9.02 [23]. Para los análisis de escala se utilizó SPSS 27.

### 5. Resultados

Respecto a la traducción del cuestionario, la evaluación realizada por jueces, concluyó en su idoneidad para el contexto cultural de nuestros alumnos. Así mismo, la prueba piloto realizada con los estudiantes, validó la claridad de los ítems. En cuanto al análisis de los datos, se realizaron pruebas previas con la finalidad de determinar si los datos eran aptos para realizar un análisis factorial. Concretamente, se comprobó la medida de adecuación maestra Kaiser-Meyer-Olkin ( $KMO=0.89$ ), por lo que concluimos que la matriz era adecuada para el análisis factorial. También comprobamos el test de esfericidad de Bartlett que fue estadísticamente significativo ( $X^2 = 2069.0 (153); p < 0.001$ ).

En el cuadro 1, se pueden ver los resultados del AFE, los datos mostraron una estructura unifactorial y parsimoniosa, el modelo propuesto se extrae en base al análisis paralelo, que aconseja extraer sólo un factor [31], de la misma manera el análisis MIREAL (0,21) sugiere que es unidimensional. Basándonos en los análisis realizados se opta por extraer un único factor, denominado "actitud hacia la programación". Se eliminó el ítem número dos debido a que tenía una saturación muy baja en el factor. Además, se han tenido que eliminar aquellos ítems que estaban muy correlacionados y/o que compartían mucha varianza residual. Específicamente se eliminaron los ítems: 5, 10, 11, 12, 16 y 17. Esto ya se esperaba porque las raíces de los ítems correspondientes estaban redactadas de manera muy similar o con un contenido similar. Como se puede ver en el Cuadro 1, todas las saturaciones son superiores a

0,3 y todos los índices de ajuste son adecuados, incluidos la raíz de la desviación cuadrática media ( $RDCM = 0,06$ ). Las medidas comparativas de ajuste en relación con el modelo nulo fueron adecuadas. De esta manera, los datos sugieren claramente que la solución propuesta es sostenible.

Respecto al índice de fiabilidad calculado para el instrumento, el alpha de Cronbach indica que la escala tiene una muy buena consistencia interna ( $\alpha = .97$ ). Así mismo, obtuvimos resultados satisfactorios al evaluar la validez del constructo de la versión española del *Computer Programming Attitude Scale*. Específicamente, obtuvimos correlaciones positivas con los tres ítems del cuestionario *ad hoc*. Ítem 1 "En general, tengo creencias positivas hacia la programación" ( $r = .21, p < .01$ ). Ítem 2 "En general, tengo sensaciones agradables hacia la programación" ( $r = .27, p < .01$ ). Ítem 3 "Habitualmente me siento cómodo/a realizando actividades relacionadas con la programación" ( $r = .25, p < .01$ ).

### 6. Discusión y conclusiones

Los resultados del presente estudio sugieren que la versión española del *Computer Programming Attitude Scale* (SV-CPAS) tiene unas buenas propiedades psicométricas. Por lo tanto, nos provee de una medida en español del constructo "actitud hacia la programación" que se describe como las diferentes actitudes de los estudiantes universitarios hacia la programación, que pueden presentarse en forma de afecto, cognición y comportamiento. El objetivo principal del presente estudio fue la adaptación y validación del cuestionario *Computer Programming Attitude Scale* (CPAS) en población universitaria española, es la primera vez que se presenta esta escala en español. Los resultados del análisis factorial exploratorio del CPAS-versión española muestran una estructura unifactorial, esto difiere de la versión original diseñada para estudiantes universitarios turcos (Cetin y Ozden, 2015), la cual presenta una estructura factorial de tres factores: afecto, cognición y comportamiento hacia la programación. Esta discrepancia se puede deber a razones culturales. Por ejemplo, se eliminó el ítem número dos "programar es una habilidad específica" porque presentó una saturación muy baja en el factor, en el contexto español este ítem no contribuye a analizar las actitudes hacia la programación, debido que una habilidad específica no describe ninguna posición hacia la programación. Por otra parte, fueron eliminados los ítems: 5, 10, 11, 12, 16 y 17 porque compartían mucha varianza residual, los ítems redundantes expresan la misma idea con una redacción mínimamente distinta, esto genera un problema cuando explican más varianza que la que es directamente explicada por el factor común.

Ítem	Factor 1
1. Creo que programar es frustrante.	-0,65
2. Programar es una habilidad específica.	-
3. Creo que programar es innecesario.	-0,48
4. La programación no tiene importancia en la vida diaria.	-0,40
5. Siempre me esfuerzo por escribir un programa mejor.	-
6. Programar es aburrido.	-0,79
7. La programación facilita la vida humana.	-0,31
8. Si encuentro un problema que no puedo resolver a corto plazo mientras programo, no me doy por vencido/a hasta que lo resuelvo.	-0,40
9. Me pongo nervioso/a mientras programo.	-0,55
10. Participo en proyectos de programación si tengo la oportunidad.	-
11. Las actividades relacionadas con la programación me ponen nervioso/a.	-
12. Busco información y recursos para ser un/a buen/a programador/a.	-
13. Me aburro mucho cuando empiezo a programar.	-0,77
14. La programación mejora mis habilidades para resolver problemas.	-0,63
15. Una vez que empiezo a trabajar en un programa, trato de terminarlo antes que nada.	-0,34
16. La sola idea de escribir código me pone nervioso/a.	-
17. Creo que muchos de los problemas que quedan sin resolver en nuestra época se resolverán gracias a los avances en programación.	-
18. Sigo los avances en programación.	-0,41
<b>Fiabilidad</b>	<b>0,93</b>

Cuadro 1: Matriz de carga obtenida en el análisis factorial exploratorio.

## 7. Trabajo futuro

Disponer de un instrumento validado nos permitirá realizar, al inicio de curso, un análisis detallado de la actitud de los alumnos ante la programación. Dicho estudio se puede analizar, de forma segmentada, teniendo en consideración aspectos como el género, estudios previos, grado que cursa el alumno, etc. A partir de las conclusiones extraídas, se pueden planificar diversas actuaciones para adaptar materiales, metodologías, tecnologías... con el objetivo de incrementar la motivación de estos alumnos hacia las asignaturas de programación.

## Referencias

- [1] Isaiah T. Awidi y Mark Paynter. "The impact of a flipped classroom approach on student learning experience". En: *Computers and Education* 128. September 2018 (2019), págs. 269-283.
- [2] Recep Çakir y Sabri Serkan Tan. "Development of educational applications on the social network of facebook and its effects on students' academic achievement". En: *Kuram ve Uygulamada Eğitim Bilimleri* 17.5 (2017), págs. 1525-1546.
- [3] Ibrahim Cetin y M. Yasar Ozden. "Development of computer programming attitude scale for university students". En: *Computer Applications in Engineering Education* (2015).
- [4] Brenda Cheang, A. Kurnia, Andrew Lim y Wee-Chong Oon. "On automated grading of programming assignments in an academic institution". En: *Computers and Education* 41.2 (2003), págs. 121-131.
- [5] Emre Çoban, Özgen Korkmaz, R. Çakir y Feray Ugur Erdogmus. "Attitudes of IT teacher candidates towards computer programming and their self-efficacy and opinions regarding to block-based programming". En: *Education and Information Technologies* (2020).
- [6] António Coelho, Enrique Kato, João Xavier y R. Gonçalves. "Serious game for introductory programming". En: *International Conference on Serious Games Development and Applications*. Springer. 2011, págs. 61-71.
- [7] Alexandre De Andrade Barbosa y Evandro De Barros Costa. "Simulated learners in peers assessment for introductory programming courses". En: vol. 1432. 2015, págs. 55-64.
- [8] Ronald Erdei, John A. Springer y David M. Whittinghill. "An impact comparison of two instructional scaffolding strategies employed in

- our programming laboratories: Employment of a supplemental teaching assistant versus employment of the pair programming methodology”. En: vol. 2017-October. 2017, págs. 1-6.
- [9] Lisa Facey-Shaw y Paul Golding. “Effects of peer tutoring and attitude on academic performance of first year introductory programming students”. En: vol. 2005. 2005, S1E-1-S1E-6.
- [10] Deniz Mertkan Gezgin, Muge Adnan y Meltem Acar Guvendir. “Mobile learning according to students of Computer Engineering and Computer Education: A comparison of attitudes”. En: *Turkish Online Journal of Distance Education* 19.1 (2018), págs. 4-17.
- [11] Anabela de Jesus Gomes, António José Mendes y Maria José Marcelino. “Computer Science Education Research: An Overview and Some Proposals”. En: *Innovative Teaching Strategies and New Learning Paradigms in Computer Programming* (2015), págs. 1-29.
- [12] Anabela de Jesus Gomes, Álvaro Nuno Santos y António José Mendes. “A study on students’ behaviours and attitudes towards learning to program”. En: *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE* (2012), págs. 132-137.
- [13] Jamie Gorson y Eleanor O’Rourke. “How do students talk about intelligence? An Investigation of Motivation, Self-efficacy, and Mindsets in Computer Science”. En: 2019, págs. 21-29.
- [14] Halit Gunbatar Mustafa Serkan abd Karalar. “Gender differences in middle school students’ attitudes and self-efficacy perceptions towards MBlock programming”. En: *European Journal of Educational Research* 7.4 (2018), págs. 925-933.
- [15] Mehmet Bans Horzum y Öziem Çakir. “The validity and reliability study of the Turkish version of the online technologies self-efficacy scale”. En: *Kuram ve Uygulamada Egitim Bilimleri* 9.3 (2009), págs. 1343-1356.
- [16] Yen-Ping Huang y Yueh-Min Huang. “Programming language learning supported by an accredited course strategy”. En: 2013, págs. 327-329.
- [17] Akpan Iniobong, Patrick Uko y Theresa Ekanem. “Computer anxiety, computer self-efficacy and attitude towards internet among secondary school students in Akwa Ibom State, Nigeria”. En: 2013, págs. 193-198.
- [18] Reyes Juárez-Ramírez, Christian X. Navarro, Verónica Tapia-Ibarra, Ricardo Macías-Olvera y César Guerra-García. “What is Programming? Putting all Together - A Set of Skills Required”. En: *2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT)*. 2018, págs. 11-20.
- [19] Mario Konecki, N. Kadoić y Rok Piltaver. “Intelligent assistant for helping students to learn programming”. En: 2015, págs. 924-928.
- [20] Özgen Korkmaz, Recep Çakir y M. Yaşar Özden. “A validity and reliability study of the computational thinking scales (CTS)”. En: *Computers in Human Behavior* 72 (2017), págs. 558-569.
- [21] Carmen Lacave, Ana Molina, M. Fernández y Miguel Redondo. “Análisis de la fiabilidad y validez de un cuestionario docente”. En: *ReVisión* 9.1 (2016), pág. 2.
- [22] Essi Lahtinen, Kirsti Ala-Mutka y Hannu-Matti Järvinen. “A study of the difficulties of novice programmers”. En: *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* (2005), págs. 14-18.
- [23] Urbano Lorenzo-Seva y Pere J. Ferrando. “FACTOR: A computer program to fit the exploratory factor analysis model”. eng. En: *Behavior research methods* 38.1 (2006), págs. 88-91.
- [24] Antonio Manso, Celio Goncalo Marques y Paulo Santos. “Algorithmi: Software system to support the learning of programming”. En: 2018.
- [25] Jordi Miró. “Translation, validation, and adaptation of an instrument to assess the information-seeking style of coping with stress: the Spanish version of the Miller Behavioral Style Scale”. En: *Personality and Individual Differences* 23.5 (1997), págs. 909-912.
- [26] Michael Newby y Darrell Fisher. “A Model of the Relationship between University Computer Laboratory Environment and Student Outcomes”. En: *Learning Environments Research* 3.1 (2000), págs. 51-66.
- [27] Hacer Özyurt y Özcan Özyurt. “Analyzing the effects of adapted flipped classroom approach on computer programming success, attitude toward programming, and programming self-efficacy”. En: *Computer Applications in Engineering Education* 26.6 (2018), págs. 2036-2046.

- [28] Muhittin Şahin y Tarık Kişla. “An analysis of university students’ attitudes towards personalized learning environments”. En: *Turkish Online Journal of Educational Technology* 15.1 (2016), págs. 1-10.
- [29] Sónia Rolland Sobral. “Two different experiments on teaching how to program with active learning methodologies: A critical analysis”. En: vol. 2020-June. 2020.
- [30] Errol Thompson, Lynn Hunt y Kinshuk Kinsluk. “Exploring learner conceptions of programming”. En: *Conferences in Research and Practice in Information Technology Series* 52 (2006), págs. 205-211.
- [31] Marieke E. Timmerman y Urbano Lorenzo-Seva. “Dimensionality assessment of ordered polytomous items with parallel analysis”. En: *Psychological Methods* 16.2 (2011). cited By 537, págs. 209-220.
- [32] Laura Toma y Jan Vahrenhold. “Self-efficacy, cognitive load, and emotional reactions in collaborative algorithms labs - A case study”. En: cited By 8. 2018, págs. 1-10.
- [33] Murat Tuncer, Yunus Doğan y Ramazan Tanaş. “Investigation of vocational high-school students’ computer anxiety”. En: *Turkish Online Journal of Educational Technology* 12.4 (2013), págs. 90-95.
- [34] R. Paul Wiegand, Anthony Bucci, Amruth N. Kumar, Jennifer L. Albert y Alessio Gaspar. “A data-driven analysis of informatively hard concepts in introductory programming”. En: *SIGCSE 2016 - Proceedings of the 47th ACM Technical Symposium on Computing Science Education* March (2016), págs. 370-375.
- [35] Mustafa Yağcı. “Web-mediated problem-based learning and computer programming: Effects of thinking style on academic achievement and attitude”. En: *Computer Applications in Engineering Education* 26.6 (2018), págs. 2012-2025.
- [36] Erman Yukselturk y Serhat Altıok. “An investigation of the effects of programming with Scratch on the preservice IT teachers’ self-efficacy perceptions and attitudes towards computer programming”. En: *British Journal of Educational Technology* 48.3 (2017), págs. 789-801.

## A. Anexo

**Escala de actitud ante la programación.** A continuación se presentan un conjunto de frases en relación a la programación. Lee cuidadosamente cada una de ellas y marca la respuesta que mejor describe tu opinión, las alternativas de respuesta para cada afirmación son: “Completamente en desacuerdo”, “En desacuerdo”, “Neutral”, “De acuerdo” y “Completamente de acuerdo” (Cuadro 2). No hay respuestas correctas ni incorrectas. Ninguna de las respuestas o resultados de los test se tendrán en cuenta en la evaluación de la asignatura. Su fin será meramente el de la investigación educativa. Todos los datos obtenidos se codificarán de forma que no aparezca ningún nombre, ni ninguna otra información que permita la identificación de los participantes. Sólo las personas relacionadas con el proyecto tendrán acceso a los archivos. Estos procedimientos están sujetos a la Ley Orgánica 15/1999 del 13 de diciembre sobre protección de datos de carácter personal. La participación en este estudio es totalmente voluntaria, por lo que el sujeto puede retirar su permiso en cualquier momento. En este caso, tus datos serían retirados de la base de datos.

	Completamente en desacuerdo	En desacuerdo	Neutro	De acuerdo	Completamente de acuerdo
1. Creo que programar es frustrante.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2. Creo que programar es innecesario.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. La programación no tiene importancia en la vida diaria.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. Programar es aburrido.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5. La programación facilita la vida humana.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6. Si encuentro un problema que no puedo resolver a corto plazo mientras programo, no me doy por vencido/a hasta que lo resuelvo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. Me pongo nervioso/a mientras programo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8. Me aburro mucho cuando empiezo a programar.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9. La programación mejora mis habilidades para resolver problemas.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10. Una vez que empiezo a trabajar en un programa, trato de terminarlo antes que nada.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11. Sigo los avances en programación.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Cuadro 2: Escala de actitud ante la programación.