

Método de evaluación del pensamiento computacional y alfabetización en código

Walkiria Helena Cordenonzi¹ , José Claudio Del Pino ,

Instituto Federal de Educación, Ciencia y Tecnología de Sul-rio-grandense (IFRSul); Universidad de Alto Taquari - Univates, Brasil

Resumen

Este artículo presenta un método de evaluación de pensamiento computacional para individuos adultos. Para lograrlo, se desarrolló toda la construcción andragógica de un curso, anclado en las teorías del aprendizaje significativo de Ausubel y de aprendizaje de adultos de Knowles. La propuesta consiste en presentar un modelo de referencia de pensamiento computacional, un método de evaluación y la clasificación de los sujetos. Como resultado de la aplicación de este método, es posible clasificar a los participantes en alfabetizado en código y pensador computacional desconectado. Este método se aplicó en un curso binacional, donde estudiantes brasileños y uruguayos comparten el mismo espacio de aprendizaje. A través de este estudio de caso y del análisis cualitativo de los resultados, el 25 % de los participantes alcanzó el estado de alfabetizado en código y el 12.5 % el de pensador computacional desconectado. Una contribución importante de este método es la combinación de instrumentos de evaluación y la clasificación resultante de los individuos, ya que no se han encontrado publicaciones similares a la fecha.

Autor de correspondencia:

¹walkiriacordenonzi@ifrsul.edu.br

Recibido: 17 de noviembre de 2020

Revisado: 30 diciembre de 2020

Aprobado: 02 de junio de 2021

Publicado: 19 de julio de 2021



Palabras clave: aprendizaje de adultos, alfabetización informacional, informática, método de evaluación

Para citar este artículo: Cordenonzi, W., & Del Pino, J. (2021). Método de evaluación del pensamiento computacional y alfabetización en código. *Praxis & Saber*, 12(31), e11750. <https://doi.org/10.19053/22160159.v12.n31.2021.11750>

Method for the assessment of computational thinking and code literacy

Abstract

This article reports a method for the assessment of computational thinking for adults. To this end, the entire andragogical construction of a course was developed, founded on Ausubel's theories of meaningful learning and on Knowles' theories of adult learning. The proposal consists of presenting a reference model of computational thinking, a method of assessment, and the classification of subjects. As a result of the application of this method, it is possible to classify participants into *code literate* and *unplugged computational thinker*. This method was implemented in a bi-national course, where Brazilian and Uruguayan students share the same space for learning. Through this case study and a qualitative analysis of the results, 25% of the participants achieved the status of *code literate* and 12.5% of the participants achieved the status of *unplugged computational thinker*. An important contribution of this method is the combination of instruments for assessment and the resulting classification of individuals, as no similar publications have been found to date.

Keywords: adult learning, information literacy, informatics, method for assessment

Método de avaliação do pensamento computacional e da alfabetização em código

Resumo

Este artigo apresenta um método para a avaliação do pensamento computacional para adultos. Para isso, toda a construção andragógica de um curso foi desenvolvida, fundado nas teorias de Ausubel sobre aprendizagem significativa e nas teorias de Knowles sobre a aprendizagem de adultos. A proposta consiste em apresentar um modelo de referência de pensamento computacional, um método de avaliação e a classificação dos sujeitos. Como resultado da aplicação deste método, é possível classificar os participantes como *alfabetizados em código* e *pensadores computacionais desplugados*. Este método foi implementado num curso binacional, onde estudantes brasileiros e uruguaios partilham o mesmo espaço de aprendizagem. Através deste estudo de caso e de uma análise qualitativa dos resultados, 25% dos participantes alcançaram o estatuto de *alfabetizados em código* e 12,5% o de *pensadores computacionais desplugados*. Uma contribuição importante deste método é a combinação de instrumentos de avaliação e a classificação resultante dos indivíduos, uma vez que até à data não foram encontradas publicações semelhantes.

Palavras-chave: educação de adultos, literacia informática, informática, método de avaliação

Muchos investigadores comparten la misma comprensión cuando se trata de *pensamiento computacional* (PC): que no hay consenso y que es una habilidad asociada con el siglo XXI. Román *et al.* (2017) agregan que aún no hay consenso sobre cómo incorporar este tipo de pensamiento en el programa escolar, ni sobre cómo evaluarlo. Para que los ciudadanos ejerzan su ciudadanía plena, su libertad y su derecho a un puesto de trabajo, la alfabetización en código es fundamental, ya que convierte al individuo en productor de tecnología y no solo en un consumidor de esta.

Wing (2017) propuso el siguiente concepto: “el pensamiento computacional está constituido por los procesos de pensamiento involucrados en la formulación de un problema y en la expresión de su(s) solución(es) de tal manera que un humano o máquina pueda ejecutarlos” (p. 8). La abstracción es la habilidad de más alto nivel dentro del proceso de PC.

La *Computer Science Teachers Association* (CSTA) recomienda algunas habilidades para desarrollar el PC: abstracción, pensamiento algorítmico, diseño, escala y reconocimiento de patrones (Yadav *et al.*, 2015). Esta asociación detalla que lo que se debe desarrollar en el PC es la capacidad de manipular representaciones simbólicas. Además, existen algunos estudios que mencionan como importantes las habilidades de abstracción y desarrollo de algoritmos, entre ellos Kalelioglu *et al.* (2016), Román (2016) y Lee *et al.* (2020). Brackmann (2017) aborda el concepto de PC desconectado, es decir, cuando se trabaja con PC sin el uso de la computadora.

En esta investigación, que tiene como objeto sujetos adultos —que ya han completado la educación básica—, se debe a que existen pocas investigaciones sobre el desarrollo de el PC en la edad adulta.

Kalelioglu *et al.* (2016) investigaron 125 artículos científicos sobre el uso de PC en diferentes niveles educativos, y concluyeron que solo el 24,8 % de estos estudios se realizaron en la educación superior. Por eso, es esencial resaltar la importancia de enfocar el PC a los adultos como medio para mejorar su desempeño profesional y personal, corroborado también por Tikva e Tambouris (2021).

En la Universidad Amasya, en un grupo de estudiantes, se aplicó una escala para determinar los niveles de *habilidad de pensamiento computacional* (Korkmaz *et al.*, 2017). Como resultado, presentaron el éxito y la confiabilidad del método propuesto. Sin embargo, no describieron cómo se aplicó el método ni cómo se llevó a cabo el desarrollo de habilidades.

En relación con las pruebas más utilizadas para evaluar el PC, Moreno *et al.* (2018) señalan el CT-Test, el Bebras International Challenge on Informatics and Computational Thinking¹ y el Dr. Scratch, que deben usarse juntos, pues son compatibles y complementarios. El primero fue diseñado por Román (2016) específicamente para alumnos españoles al final del ciclo básico y secundario y mide el nivel de desarrollo del PC a través de un cuestionario. Él llegó a la conclusión de que el PC es una habilidad cognitiva para resolver problemas. El Dr. Scratch² es una herramienta web que analiza automáticamente los proyectos desarrollados en Scratch. Además, el Bebras es un desafío que consiste en preguntas —o tareas— que tiene como objetivo evaluar el nivel de desarrollo de las habilidades del PC (Dagiené & Futschek,

1 Disponible en (<https://www.bebas.org/>)

2 Disponible en (<http://www.drscratch.org/>).

2008), y puede emplearse en cualquier edad (Moreno *et al.*, 2018). Algunas preguntas de los test de este trabajo fueron tomadas de Bebras y traducidas al portugués y al español.

En resumen, el campo de evaluación del PC todavía carece de investigaciones. Fue posible analizar que la mayoría de las metodologías propuestas son cualitativas (Raabe *et al.*, 2017) y que existen pocas herramientas de evaluación que produzcan retroalimentación inmediata y se limiten al entorno de desarrollo utilizado.

La investigación presentada en este artículo tiene como objetivo comprender y analizar el desarrollo del PC en sujetos adultos, formados en escuelas brasileñas y uruguayas. Para esto, se desarrolló un *modelo de referencia de pensamiento computacional* (MRPC) y un conjunto de materiales instruccionales para evaluar el desarrollo del PC. También se presenta el resultado de su aplicación en un caso piloto, que involucra a sujetos de ambas nacionalidades.

Estas propuestas se basaron en la teoría del aprendizaje significativo de Ausubel y en la andragogía de Knowles. Para Ausubel *et al.* (1983), el sujeto aprende cuando ancla nuevos conocimientos en su estructura cognitiva. El factor más importante en el aprendizaje son los conocimientos previos del alumno: las estructuras cognitivas que ya tiene el aprendiz y que permitirán la conexión de nuevos conocimientos. Ausubel resume su teoría en la siguiente frase: “El factor más importante que influye en el aprendizaje es lo que el alumno ya sabe. Descúbrelo y enséñalo en consecuencia” (Ausubel *et al.*, 1983, p. 1).

Knowles (1981) definió la andragogía “como el arte y la ciencia de ayudar a los adultos a aprender” (p. 43). Enseñar a los adultos es diferente a enseñar a los niños. El docente andragógico debe considerar que el proceso de aprendizaje se centra en el alumno, quien participa en la elaboración del programa —contenido— por trabajar y se da cuenta de su aplicación práctica.

Procedimientos metodológicos

Caracterización de la investigación

Una investigación científica comienza cuando hay un problema que se puede resolver (Gil, 2002). Así, a partir de la pregunta descriptiva del problema para investigar, se formularon las hipótesis y la investigación tomó forma para llevarse a cabo. Triviños (2015) agrega que el problema de investigación es un escenario o contexto que necesita ser aclarado: en este caso, desde la perspectiva de los adultos, hay pocos trabajos publicados en el campo del PC anclados en dos teorías de aprendizaje: aprendizaje significativo y andragogía.

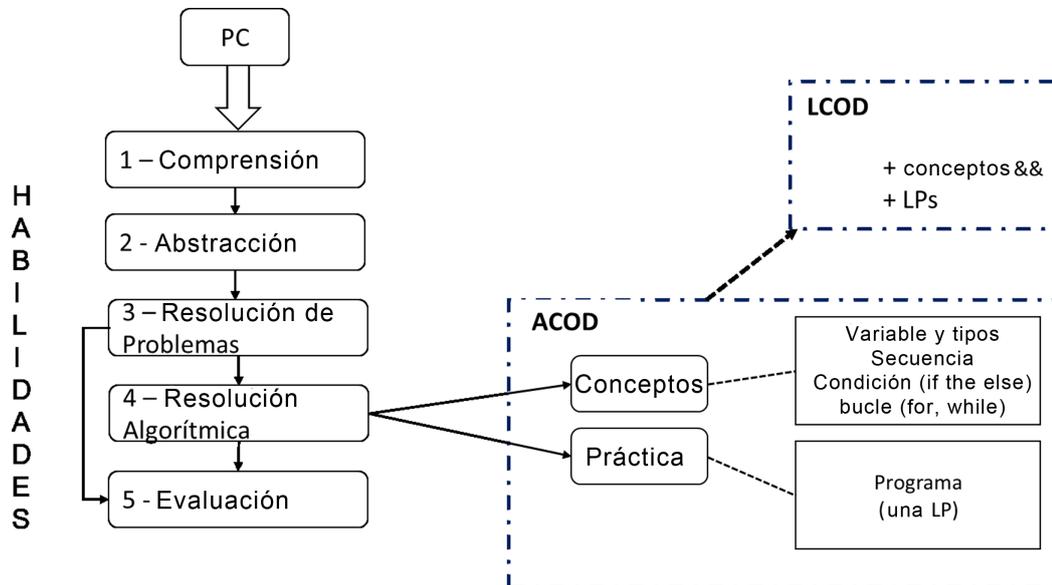
Este trabajo se realizó según el método inductivo, con estudio de caso y evaluación de resultados de forma cualitativa y cuantitativa. Los documentos exigidos por la Resolución N°466/2012 (Conselho Nacional de Saúde) de Brasil, como el *Formulario de Consentimiento Libre y Aclarado* y la *Carta de Consentimiento de la Institución*, también fueron desarrollados y firmados. Los integrantes de la investigación son los estudiantes de los cursos binacionales, que incluyen estudiantes brasileños y uruguayos y sus diferentes formaciones. Por lo tanto, las lenguas maternas en estos cursos son el portugués y el español (Cordenonzi *et al.*, 2014).

Se propuso un MRPC (figura 1), junto con un conjunto de materiales de instrucción para llevar a cabo el curso y varios tipos de test. Esto forma el constructo pedagógico para

la evaluación del PC en el sujeto, quien es clasificado como: *alfabetizado en código*, *pensador computacional desconectado* o ninguno de los anteriores.

Figura 1

Modelo MRPC



Nota. Tomado de *O desenvolvimento do pensamento computacional e as evidências da alfabetização em código em adultos* [Tesis Doctoral no publicada] de W.Cordenonzi, 2020, Universidade do Vale do Taquari - Univates.

Se considera que las siguientes son las habilidades mínimas y necesarias que se deben presentar para que el individuo pueda resolver un problema y probarlo:

1. Comprensión. Este es el primer paso para resolver un problema. El sujeto necesita entender lo que se está solicitando y en qué contexto, para poder significar y aplicar, es decir, una organización jerárquica entre la comprensión con la capacidad de abstracción. Como resultado de la falta de comprensión, el individuo no puede resolver el problema porque, a partir de la lectura, no puede identificar cuáles son las partes importantes —abstracción—. Por lo tanto, la comprensión pasa por un aprendizaje significativo, en el cual el sujeto desarrolla su conocimiento. Sus procesos cognitivos conducen a la resolución exitosa del problema, al usar sus conocimientos previos, que para Ausubel (2003) son el “factor determinante del proceso de aprendizaje” (2003, p. 7). Además, es necesario que el sujeto encuentre la relación entre los conocimientos previos y los utilice para el siguiente paso, en el que debe lograr resolver el problema. En síntesis, la comprensión es una habilidad innovadora del sujeto frente a la resolución de un problema, que es uno de los factores críticos para el desarrollo del PC.

2. Abstracción. Es un proceso para decidir qué detalles deben ser destacados y cuáles descartar. Wing (2006) complementa que la abstracción es el proceso de más alto nivel cuando se piensa en PC. En este punto, no estamos de acuerdo con dicho autor, pues creemos que, para que el sujeto lleve a cabo el proceso de abstracción, primero debe haber entendido el problema para luego realizar el proceso de abstracción.

3. Resolución de problemas. Es la capacidad del sujeto para encontrar una o varias soluciones, independientemente del formato de su respuesta. Del mismo modo, y al justificar su importancia, Ausubel *et al.* (1983) describieron que la capacidad de resolver problemas es el objetivo principal de la educación (p. 448).

4. Resolución algorítmica. Los algoritmos son una secuencia de pasos lógicos, descritos en cualquier lenguaje. Los programas son la “traducción” de los algoritmos a un lenguaje de programación. Cuando el sujeto encuentra una o más soluciones al problema y se vuelve capaz de expresar la solución a través de un algoritmo, está desarrollando esta habilidad.

5. Evaluación. Es la última habilidad propuesta, que se entiende como la capacidad del sujeto después de presentar la resolución del problema a través de un algoritmo y/o programa, para probar y evaluar su corrección. En otras palabras, es entender lo que se ha implementado. Es la capacidad de analizar un código fuente o ejecutable en cuanto a sus salidas.

Para ejecutar dicha práctica con los alumnos, fue elegida la *app* Inventor³, pues, desde una interfaz simple e intuitiva, el usuario es capaz de desarrollar sus aplicaciones en poco tiempo. Ya en los primeros encuentros, el alumno logra desarrollar una aplicación simple, para el sistema operativo Android —que posee la mayoría de los usuarios—. Además, al elegir los temas por desarrollar, se evaluaron aspectos como la motivación del aprendiz —por ejemplo, mejores oportunidades laborales, salarios más altos y satisfacción personal—. Con base en la elección de los temas de las *apps*, el modelo andragógico de Knowles fue una de las principales referencias, ya que fomenta en el aprendiz la necesidad del conocimiento.

Cuando el alumno termina su *app*, la envía al profesor, quien realiza la corrección y hace un *feedback*, porque “sabemos que la andragogía debe ser más personalizada para adecuarse a la singularidad entre los adultos” (Knowles *et al.*, 2011, p. 146), ya que el enfoque del aprendizaje está en el alumno y el profesor/tutor ha de acompañar este proceso.

Con el propósito de lograr dicho acompañamiento, se utiliza un *ambiente virtual de aprendizaje* (AVA) que también permite la implementación de la comunidad de práctica (CdP). En esta propuesta, la comunidad puede ser operacionalizada a través de un foro, con interacción asincrónica. La misma interacción también puede ocurrir en una herramienta sincrónica, la cual puede ser implementada creando un grupo en WhatsApp.

A continuación, se definió el documento *Plan de Clases*, que contiene toda la información del curso, así como cada encuentro y cada aplicación que será construida. Cuenta con la definición de los objetivos, temas, recursos utilizados, entre otros.

Al principio del curso, se sugiere que el profesor presente el contrato de aprendizaje a los alumnos. Este documento sirve para organizar y guiar los objetivos del proceso de enseñanza y aprendizaje, para todos los involucrados, al fomentar la autonomía del aprendiz (Knowles *et al.*, 2011). Cabe señalar que el profesor presenta este contrato a los alumnos, que pueden contribuir en él, de acuerdo con sus necesidades.

Método de evaluación del PC

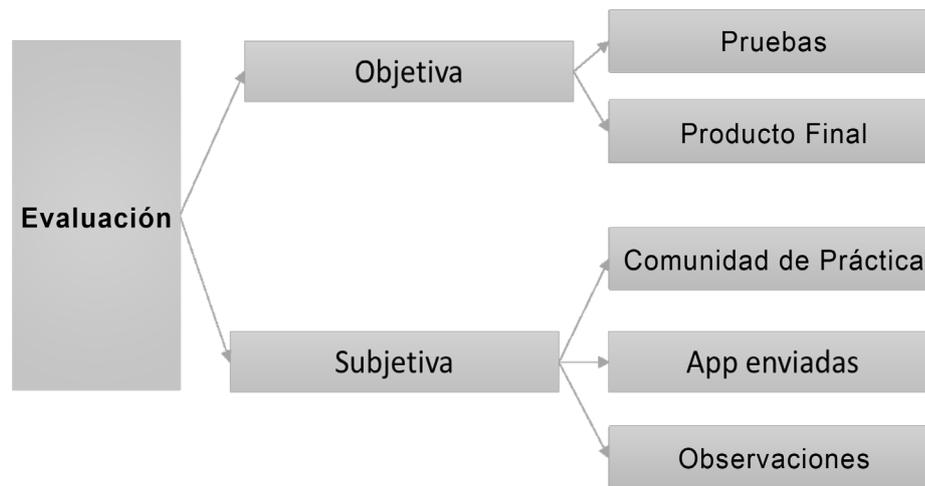
Como evaluar no es un proceso simple, se proponen varios instrumentos para lograr este

³ Herramienta de creación de *apps*, ofrecida gratuitamente en la internet por el MIT en <http://ai2.appinventor.mit.edu>.

fin, los cuales se agrupan en dos conjuntos: instrumentos objetivos y subjetivos (figura 2). Los instrumentos objetivos están compuestos por pruebas —pre-, post- e intermedias— y por el producto final, con sus respectivos indicadores. Los subjetivos son: observación, CdP y *apps* desarrolladas durante el curso. El término *CdP* fue propuesto por Wenger (2002), quien lo define como un grupo de personas que comparten una pasión sobre algo que saben hacer y que interactúan para aprender cómo hacerlo mejor; es generar y compartir conocimiento. Enfatiza que las CdP apuntan a resolver problemas, ya que todos los que participan aprenden. Esta participación debe ser efectiva.

Figura 2

Tipos de instrumentos de evaluación



Nota. Tomado de *O desenvolvimento do pensamento computacional e as evidências da alfabetização em código em adultos* [Tesis Doctoral no publicada] de W.Cordenonzi, 2020, Universidade do Vale do Taquari - Univates

Después de realizar todas las pruebas, de entregar el producto final —*app*— y de evaluarlo adecuadamente, se resumen las preguntas de la prueba. El resultado deberá alcanzar un valor mínimo del 60 % de respuestas correctas. Este valor también debe observarse para los indicadores del producto final. Con estos valores conocidos, se puede sugerir que el individuo está *alfabetizado en código* (ACod).

- Un individuo ACod es aquel capaz de leer, interpretar y escribir código fuente y generar código ejecutable —programa—.
- Un individuo *letrado en código* (LCod) es un ACod que extrapola sus habilidades y competencias de código en diferentes lenguajes de programación (Cordenonzi et al., 2020, p. 150).

En la evaluación objetiva, se proponen la prueba previa —o prueba 1—, una prueba intermedia y una última —prueba posterior—. Las preguntas que conforman las dos primeras pruebas fueron tomadas de Bebras y, por lo tanto, ya estaban validadas. La prueba posterior se desarrolla teniendo en cuenta todas las habilidades, con interpretación de código —habilidad 5—. Todas las preguntas son de opción múltiple. Una de las alternativas es la opción “no entendí la pregunta”, con el fin de evaluar la habilidad 1 —comprensión—. En total se proponen nueve preguntas. Para cada pregunta se conoce(n) la(s) habilidad(es)

que se vinculan, siempre en ambos idiomas. En la figura 3, se puede apreciar una pregunta de la prueba posterior.

Figura 3

Ejemplo de pregunta de la prueba posterior

<p>Na interface proposta, é possível eliminar o botão de Sexo (Homem e Mulher), sem alterar a funcionalidade? (H 1, 2, 3, 4, 5)</p> <p>a) <u>não entendi a questão</u></p> <p>b) <u>sim</u></p> <p>c) <u>não</u></p> <p>d) <u>em partes</u></p>	<p>En la interfaz propuesta, ¿es posible quitar las teclas de Sexo (Mujer y Hombre) sin cambiar la funcionalidad del aplicativo?</p> <p>a. <u>no entendí la pregunta</u></p> <p>b. <u>sí</u></p> <p>c. <u>no</u></p> <p>d. <u>parcialmente</u></p>
--	---

Nota. Tomado de *O desenvolvimento do pensamento computacional e as evidências da alfabetização em código em adultos* [Tesis Doctoral no publicada] de W.Cordenonzi, 2020, Universidade do Vale do Taquari - Univates

Para evaluar el resultado de la aplicación propuesta en el proyecto final del curso, se investigó la norma ISO/IEC 25010: 2011 —Modelos de Calidad de Sistemas y Software—. Este estándar define ocho características de calidad del software y sus subcaracterísticas que se presentan para determinar el alcance de cada medida. Las utilizadas en este trabajo son: integridad, corrección, adecuación, idoneidad, aprendizaje, operabilidad, protección contra errores, estética y comprobabilidad. Para cada uno de ellos, se propusieron indicadores de presencia en la *app*. Los criterios definidos están en la tabla 1.

Tabla 1

Evaluación del proyecto final

Característica	Subcaracterística	Pregunta	¿Se encuentra presente?	
			Sí	No
Funcionalidad	Integridad	¿Hiciste todo lo planeado?		
	Corrección	¿Hace lo que se propuso correctamente?		
	Adecuación	¿Propone hacer lo que sea apropiado?		
Usabilidad	Verificar la adecuación	¿Cumplió con sus necesidades?		
	Aprendizaje	¿Es fácil de aprender?		
	Operabilidad	¿Es fácil de operar y controlar?		
	Protección contra errores	¿Tiene validación de datos de entrada?		
Manutenibilidad	Estética	¿Es agradable la interfaz?		
	Comprobabilidad	¿Es fácil de probar?		
Total				

Nota. Estas características están basadas en la ISO/IEC 25040.

Nota. Tomado de *O desenvolvimento do pensamento computacional e as evidências da alfabetização em código em adultos* [Tesis Doctoral no publicada] de W.Cordenonzi, 2020, Universidade do Vale do Taquari - Univates.

Al final de esta evaluación, el profesor deberá contar la presencia de los indicadores, en un total de 10. Al menos 6 deberán estar presentes —60 %—. Son obligatorios todos los de la característica *funcionalidad*. A partir de este resultado, el paso 5 será emitir un parecer o una nota con el fin de concluir la evaluación. Al terminar este proceso, ambas evaluaciones objetivas deberán alcanzar 60 % para que el alumno sea considerado ACod.

Para los instrumentos subjetivos, el profesor debe evaluar individualmente a cada alumno. Entonces podrá determinar si el alumno ha desarrollado todas las habilidades propuestas en el MRPC, junto con las observaciones realizadas a lo largo del curso y en la CdP. Si el estudiante no ha desarrollado adecuadamente la habilidad 4 —resolución algorítmica—, entonces el tutor puede sugerir que el estudiante ha desarrollado el PC de una manera desenchufada.

La observación directa es una evidencia que debe ser realizada por el profesor durante el transcurso de sus clases. Esto significa observar a sus alumnos en tiempo real e intentar reflejar y juzgar la posición del alumno de acuerdo con el contexto y con el contenido trabajado. Durante las clases, el estudiante debe ser observado. Cuando una *app* se finalice, o incluso al final de cada encuentro, el profesor solicita que la *app* esté disponible, tanto para él como para el grupo de colegas. A partir del código fuente, el profesor podrá acompañar la producción del alumno en relación con el software desarrollado, verificar la integridad, si hubo errores, entre otros. Por consiguiente, al proporcionar un *feedback* adecuado y analizar el código fuente, se puede percibir la presencia —o no— de las habilidades referentes a aquella *app*. De esta manera, el profesor se vuelve capaz de dibujar el perfil de conocimiento de su alumno, de forma individual. Aun en el sesgo de subjetividad—con la observación y el análisis de desarrollo de cada alumno—, corresponde al profesor incentivar el uso de la CdP, para que el grupo comparta sus experiencias —como ya se vio en el acompañamiento de las *apps*—, sus dudas y sus aciertos. A partir de las interacciones de los alumnos, el profesor puede observar estas comunicaciones para componer el perfil del alumno con respecto al desarrollo de su PC, a través de las habilidades sugeridas en el MRPC. A partir de estos instrumentos, en los cuales la actuación del profesor es fundamental, le corresponde a él juzgar el conocimiento y el desarrollo del PC del alumno.

Por un lado, anclados en Ausubel y Knowles y, por otro lado, en la comprensión y evaluación del PC, se propusieron estos constructos pedagógicos, que tratan del desarrollo de habilidades para la resolución de problemas en la forma de programas —o no— y para evaluar este proceso. El resultado deberá mostrar indicios de las habilidades propuestas con el fin de afirmar que el sujeto se ha convertido en un ACod, o ha desarrollado las habilidades, excepto la programación —desconectado—.

Caso piloto

La primera experiencia realizada fue con el curso binacional de Técnico Terciario. Se planearon once encuentros. Durante el primero, los estudiantes se mostraron receptivos y entusiasmados con la propuesta del curso ¡Yo Programa 1.0!. Además, se presentó un contrato de aprendizaje, como sugirió Knowles, a los estudiantes, quienes estuvieron de acuerdo con su contenido.

Se creó un grupo en WhatsApp y se pusieron a disposición en el AVA foros con el

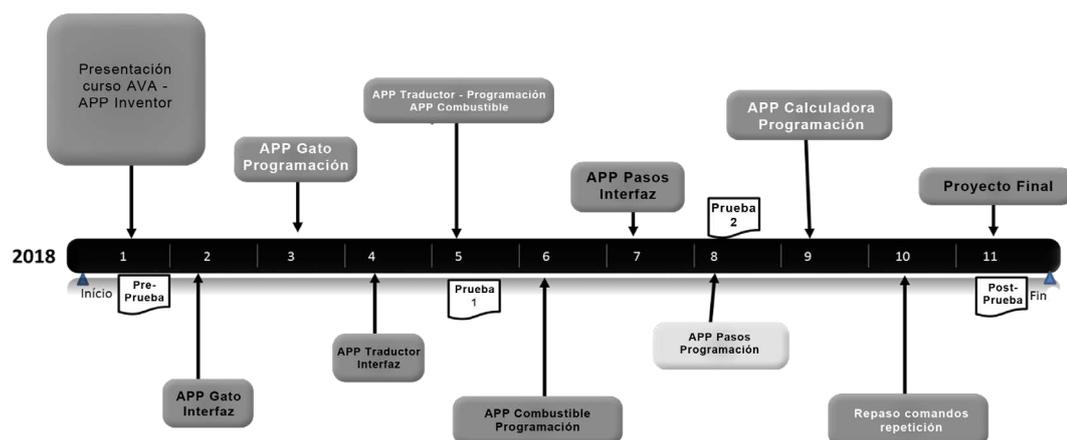
objetivo de incentivar la interacción entre los alumnos y el profesor para fomentar el trabajo colaborativo. Estos dos mecanismos de interacción tienen como objetivo la implementación de la CdP del tema, que es el enfoque del curso.

En el undécimo encuentro, los alumnos presentaron la prueba posterior —o prueba final—, en las versiones en portugués y español. Para que la capacidad de comprender el problema no se viera comprometida por el entendimiento —y traducción— del idioma, cada alumno eligió el idioma en que le gustaría responder. Todos optaron por su lengua materna. También en esta clase se explicó el proyecto final. Entre todos discutieron el tema del trabajo, el interés para su profesión, la forma de evaluación y el tiempo para el desarrollo. Luego se escribieron las especificaciones del proyecto final, que consistió en realizar un cálculo simplificado de fletes por carretera. Este documento está disponible en el AVA, en el que fueron descritos todos los datos necesarios.

La última reunión fue dedicada a la comprensión del proyecto final. Es decir, la habilidad de comprensión fue una de las más trabajadas. Se cree que la mayor dificultad de los individuos aún reside en la comprensión —leer un problema y entender lo que está escrito—, ya que sin esta habilidad todas las demás quedan comprometidas. Además, se pudo comprobar que, después de la explicación, del debate y de la interacción con la investigadora y entre los colegas, los alumnos lograron entender el problema del flete. Las demás habilidades, como la abstracción, fueron empleadas y desarrolladas de manera que se percibiera la evolución de su aprendizaje. Surgieron algunas dudas en la lectura de las variables, en el sentido de confirmar su conocimiento y la exactitud de lo que deberían programar. A partir de esas preguntas, se percibió que se había entendido la habilidad de resolución algorítmica. El resumen de los encuentros se presenta en la figura 4.

Figura 4

Línea temporal de las actividades del curso



Nota. Tomado de *O desenvolvimento do pensamento computacional e as evidências da alfabetização em código em adultos* [Tesis Doctoral no publicada] de W.Cordenonzi, 2020, Universidade do Vale do Taquari - Univates

En la primera clase se les solicitó a los alumnos que respondieran el cuestionario *Conociéndote*. En el quinto encuentro fue aplicada la *Prueba 1* y en el octavo encuentro se presentó la *Prueba 2*. Ambas se pusieron a disposición a través de las herramientas de

comunicación del grupo: el AVA *Foro y Tareas* y en el grupo de WhatsApp.

Para completar el cronograma del proyecto final, se acordó un tiempo con los alumnos, quienes publicaron el archivo correspondiente en el entorno virtual. Un alumno envió el proyecto por este medio y los demás lo enviaron por correo electrónico a la profesora.

Resultados y discusión

El grupo inicial contó con diez alumnos, dos de ellos brasileños. El primer contacto para comprender el contexto de los estudiantes fue a través de las preguntas del cuestionario *Conociéndote*. Solamente nueve alumnos respondieron las preguntas, aunque el cuestionario estuvo disponible en el AVA durante todo el curso.

La respuesta fue unánime sobre el desconocimiento al programar. Sin embargo, apenas dos alumnos no pudieron responder qué es un algoritmo y solo dos respuestas se completaron parcialmente. Cuando se les preguntó sobre qué son las variables y los comandos de condición de repetición, el 44 % no respondió. Dos alumnos relacionaron el concepto de variables con la resolución de ecuaciones matemáticas. Esto se evidencia en la frase: “variable: símbolo o valor numérico que conforma una ecuación matemática”. Los demás no respondieron correctamente. Solamente un alumno respondió a todas las preguntas de este tema. Esto muestra que no hay subsunción⁴ para el conocimiento sobre la habilidad de resolución algorítmica.

Cuando se le preguntó sobre el enunciado “programar o ser programado”, es decir, si el sujeto desarrolla su software para satisfacer sus necesidades específicas o deja que otras personas lo hagan, más de la mitad de los encuestados coincidió en la importancia del conocimiento sobre programación. Un alumno complementó al decir: “saber programar para adaptarse a las condiciones y exigencias del siglo XXI, y no ser siempre programados”. Por más que el grupo haya dividido sus opiniones sobre este tema, el 100 % de los alumnos está de acuerdo en que el mercado exige cada vez más este conocimiento. También son unánimes con relación a las expectativas del curso: aprender, dada la importancia que este conocimiento exige en el mercado actual. Esto se puede corroborar con las conclusiones de los alumnos: “poder aprender más sobre el ámbito informático y adquirir conocimientos que me aporten en el ámbito laboral” y “*adquirir conhecimento de fazer aplicativos úteis para todo tipo de usuario* [adquirir conocimiento para hacer *apps* útiles para todo tipo de usuario]”. A partir de estas declaraciones, es posible considerar que los alumnos entendieron la importancia y el propósito del curso.

Durante las clases, siempre se les pedía a los alumnos que enviaran el diseño de sus aplicaciones, incluso si no estaban completas o si tenían errores. Todos los proyectos enviados fueron corregidos y devueltos a los alumnos con las debidas observaciones y correcciones en el propio código.

Con respecto a las evaluaciones, la tabla 2 muestra el resultado de las respuestas correctas por alumno con respecto a las habilidades desarrolladas.

Tabla 2

Resultado de las respuestas correctas de la evaluación objetiva

⁴ La *subsunción* es un proceso mediante el cual una idea nueva se relaciona con otra que ya existe, modificándola y estableciendo más sentido (Ausubel et al., 1983)

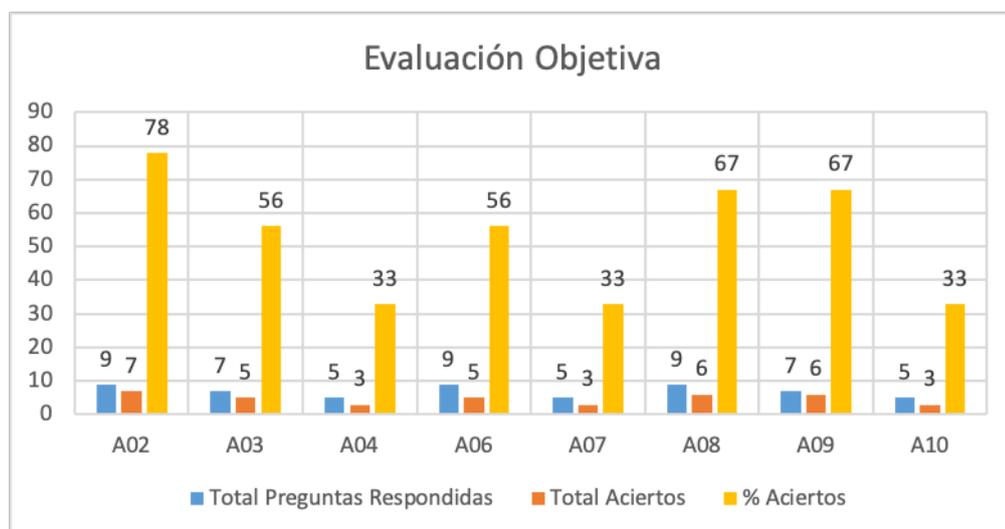
ID alumno	TT	TQ	H1/5	H2/6	H3/6	H4/4	H5/4	Total	%
A01	0	0							
A02	3	9	3	4	3	1	4	7	78
A03	2	7	3	3	2	0	0	5	56
A04	1	5	3	2	1	0	0	3	33
A05	0	0							
A06	3	9	3	4	4	1	1	5	56
A07	1	5	3	3	1	0	0	3	33
A08	3	9	3	3	4	1	1	6	67
A09	2	7	4	4	4	1	1	6	67
A10	1	5	1	2	1	0	0	3	33

Nota. Se realizaron tres pruebas, la columna TT contiene la cantidad de pruebas que el alumno respondió. La columna TQ muestra el número de preguntas respondidas por el alumno, pues se presentaron un total de nueve preguntas divididas en tres pruebas. La columna H1/5 indica que cinco preguntas —del total de 9— fueron evaluadas para la habilidad 1 —comprensión—. Y así sucesivamente en las demás habilidades. De esta manera, H2 es abstracción, H3 es resolución de problemas, H4 es la habilidad de resolución algorítmica y H5 es evaluación. En la columna Total, está el número de preguntas respondidas correctamente por el alumno y su porcentaje correspondiente en la siguiente columna. Tomado de *O desenvolvimento do pensamento computacional e as evidências da alfabetização em código em adultos* [Tesis Doctoral no publicada] de W.Cordenonzi, 2020, Universidade do Vale do Taquari - Univates

La tabla 3 muestra el resultado ya presentado con respecto a las pruebas más la evaluación de los indicadores referentes al proyecto final, con el propósito de consolidar la evaluación objetiva.

Gráfico 1

Evaluación objetiva



Nota. Tomado de *O desenvolvimento do pensamento computacional e as evidências da alfabetização em código em adultos* [Tesis Doctoral no publicada] de W.Cordenonzi, 2020, Universidade do Vale do Taquari - Univates

En el gráfico 1, se puede ver en las respuestas de los alumnos 4, 7 y 10 —quienes respondieron solo cinco preguntas—, que todos obtuvieron solo tres respuestas correctas, lo que significa que participaron solo en la prueba posterior. Por lo tanto, es posible inferir que los encuestados no habían participado en pruebas anteriores y que tampoco respondieron correctamente las preguntas 4 y 5 —más complejas—. Sin embargo, a pesar de estos resultados, el 33 % de los alumnos obtuvo más del 60 % de respuestas correctas.

Tabla 3

Resultado de la ACod

ID del alumno	% de respuestas correctas pruebas	Cantidad indicadores ProdF	% de indicadores ProdF	Parecer
A02	78	7	78	Indicios de que está ACod.
A03	56	No envió		Sin el proyecto final, no fue posible evaluar la alfabetización en código.
A04	33	4	57	No fue suficiente para evaluar la alfabetización en código.
A06	56	7	78	Indicios de que está ACod. A pesar de que el porcentaje de respuestas correctas no alcanzó el 60 %, demostró conocimientos en el proyecto final.
A07	33	No envió		No fue suficiente para evaluar la alfabetización en código.
A08	67	6	67	Indicios de que está ACod. A pesar de que el porcentaje de respuestas correctas no alcanzó el 60 %, demostró conocimientos en el proyecto final.
A09	67	7	78	Indicios de que está ACod. A pesar de que el porcentaje de respuestas correctas no alcanzó el 60 %, demostró conocimientos en el proyecto final.
A10	33	4	44	No fue suficiente para evaluar la alfabetización en código.

Nota. Tomado de *O desenvolvimento do pensamento computacional e as evidências da alfabetização em código em adultos* [Tesis Doctoral no publicada] de W.Cordenonzi, 2020, Universidade do Vale do Taquari - Univates

A partir de los datos presentados en la tabla 3, con respecto a los alumnos que no enviaron el proyecto final, la evaluación no fue posible. Por lo tanto, en cuanto a la afirmación de ser o no ACod, sus conocimientos no pueden juzgarse por falta de indicadores necesarios. Con respecto a la evaluación de los alumnos 4 y 10, no lograron alcanzar el mínimo de conocimiento. Por lo tanto, no fueron considerados ACod. En total, 4 alumnos presentaron indicios de ser ACod.

Los alumnos declararon que estaban al final del semestre y terminando muchas actividades de otras materias, por lo que un prototipo de interfaz fue desarrollado por la autora y puesto a disposición de los alumnos. Esto corrobora la posición de Santos (2010):

La andragogía resalta que los adultos tienden a tener una gran cantidad de preocupaciones y problemas que resolver fuera de la situación de aprendizaje. Por lo tanto, las demandas de la experiencia de aprendizaje no deben ser poco realistas, debe haber un equilibrio adecuado

entre el tiempo requerido para presentar la situación de aprendizaje y el tiempo requerido para lograr el aprendizaje. (p. 7)

Nuevamente se pudo percibir que la mayor dificultad radica en el primer pilar: la comprensión.

En la tabla 4, se presenta el resumen de la segunda parte de la evaluación —la subjetiva— donde aún se considera a los alumnos que se muestran en la tabla 3.

Tabla 4

Evaluación subjetiva

ID alumno	Evaluación objetiva / producto final	Apps enviadas /4	CdP	Observación
A02	78 % 78 %	3	Sí, con contribuciones importantes	A pesar de faltar a tres clases, hacía las tareas y seguía los tutoriales.
A03	56 % 0 %	0	Pocas intervenciones	Tres faltas. No lograba realizar las actividades sola.
A04	33 % 57 %	2	Sí, con contribuciones	Desarrollaba y seguía bien las actividades de la clase.
A06	56 % 78 %	3	Poca interacción. Su contacto más en lo privado	Desarrollaba y seguía bien las actividades de la clase, y en las faltas lograba seguir los tutoriales.
A07	33 % 0 %	0	Poca interacción.	Desarrollaba y seguía bien las actividades de la clase, con cierta independencia.
A08	67 % 67 %	2	Sí, con contribuciones importantes	Desarrollaba y seguía bien las actividades de la clase.
A09	67 % 78 %	4	Sí, con contribuciones importantes	Desarrollaba y seguía bien las actividades de la clase, y en las faltas lograba seguir los tutoriales.
A10	33 % 44 %	3	No participó. Solamente una vez en privado.	No lograba seguir las actividades de la clase. Las apps enviadas siempre estaban inacabadas.

Nota. Tomado de *O desenvolvimento do pensamento computacional e as evidências da alfabetização em código em adultos* [Tesis Doctoral no publicada] de W.Cordenonzi, 2020, Universidade do Vale do Taquari - Univates

A partir de los datos presentados en la tabla 4, es posible seguir el desarrollo de los alumnos con respecto a las actividades propuestas durante el curso. Es importante destacar que los estudiantes 2, 6, 8 y 9, considerados ACod desde la evaluación objetiva, presentan un patrón en la evaluación subjetiva. En otras palabras, interactuaron en la CdP, tanto con la formulación de preguntas como con respuestas a preguntas de los colegas. Además, desarrollaron las aplicaciones, tanto en clase como solos —autodidactas— y las compartieron con el grupo.

Al analizar al alumno 3, se puede notar que faltó a clases. Aunque obtuvo un puntaje en la última prueba, no logró seguir al grupo, ni aprovechó la CdP para ayudar, ni realizó el proyecto final. Por lo tanto, no se puede afirmar si hubo una mejora o una construcción con respecto al desarrollo del PC, pues su participación y desempeño no fueron suficientes. Se puede hacer un análisis similar sobre el alumno 10, que, si bien desarrolló y entregó el proyecto final, fue

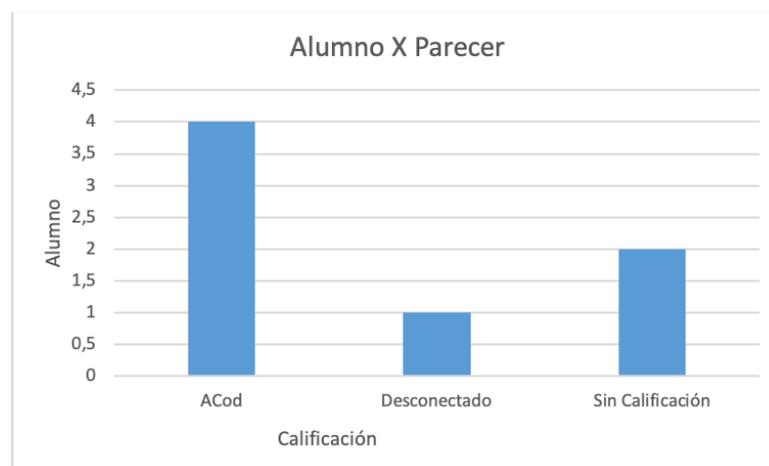
insuficiente en los indicadores. Es decir, ni siquiera desarrolló la mitad de lo especificado en el proyecto. Así mismo, se determina una conclusión análoga para el alumno 7.

Al analizar el alumno 4, este no alcanzó ACod, pero desarrolló bien las aplicaciones de la clase. Aunque solo entregó dos, se notó un interés en el tema y tuvo éxito en las primeras tres habilidades evaluadas. Se puede afirmar que este participante reúne indicios del desarrollo del PC, pero de manera desconectada.

En resumen, está claro que la unión de evaluaciones —objetiva y subjetiva— se complementan entre sí, incluso para validar a los alumnos considerados ACod. El resumen se puede ver en el gráfico 2.

Gráfico 2

Cantidad de alumnos por parecer



Nota. Tomado de *O desenvolvimento do pensamento computacional e as evidências da alfabetização em código em adultos* [Tesis Doctoral no publicada] de W.Cordenonzi, 2020, Universidade do Vale do Taquari - Univates

Este gráfico muestra el número de alumnos que fueron clasificados como ACod, en este caso, 4. De este número, apenas un alumno recibió capacitación en Brasil. El participante que desarrolló el PC de forma desconectada recibió capacitación en Uruguay y el resto no pudo ser evaluado.

Otro punto importante para la evaluación son las *apps*, las cuales, desarrolladas en clase, que al final son enviadas para la CdP de práctica y evaluadas por el profesor —que siempre hace un *feedback*—, son importantes para el aprendizaje y, en consecuencia, para el desarrollo de las habilidades mencionadas. Se puede validar que los estudiantes que lograron la ACod entregaron la mayoría de las aplicaciones. Por lo tanto, esto pone en evidencia la importancia de esa práctica en el proceso de aprendizaje.

Diez alumnos comenzaron el curso y solamente ocho llegaron al final. De este grupo, cuatro alumnos —uno brasileño— pudieron considerarse ACod. Ninguno de estos alumnos declaró conocimientos previos de programación, lograron un porcentaje mayor o igual a 60 % en las pruebas realizadas y lograron desarrollar las aplicaciones, incluso cuando no estaban en clase —solamente a través del material de instrucción proporcionado—. Además, interactuaron en la CdP al enviar lo que se solicitaba y al expresar sus dudas.

Se concluye que la CdP, a partir de su papel en el proceso de evaluación, es un elemento importante para desarrollar las habilidades mencionadas. En definitiva, los estudiantes que se comprometieron con las actividades propuestas lograron la ACod.

No hubo diferencias significativas en el desarrollo de habilidades de PC entre alumnos brasileños y uruguayos durante las interacciones en el aula. Uno de los factores que impactaron los resultados de las evaluaciones de los estudiantes fueron las ausencias. De los once encuentros realizados, las ausencias estudiantiles promediaron el 30 %. Los estudiantes clasificados como ACod, incluso sin estar presentes en clase, lograban realizar las actividades. Los demás alumnos no intentaron recuperar los contenidos perdidos. En cada encuentro se desarrolló la interfaz o la programación de una *app*. De esta manera, el alumno ya no seguía la clase, y cada problema propuesto —*app*— fue planeado para retomar los conceptos anteriores y avanzar en uno nuevo.

Cabe destacar que los alumnos no presentaban ninguna subsunción sobre algoritmos, lógica de programación ni lenguaje de programación. Esto significa comprender que, al menos, la habilidad 4 fue construida totalmente durante el curso. Todos los ítems de evaluación son importantes y complementarios. Los estudiantes que no completaron todas las tareas propuestas fueron descartados.

Conclusiones

En este artículo se presentó un método de evaluación de PC, basado en las teorías de Ausubel y Knowles, a partir de una construcción pedagógica que aborda tanto el desarrollo de habilidades para resolver problemas en forma de programas —o no— y para evaluar este proceso. El resultado muestra indicios de las habilidades propuestas, con la finalidad de afirmar que el sujeto se ha convertido en un ACod, o desarrolló las habilidades excepto la de programación —desconectado—. Ciertamente, el proceso propuesto considera las recomendaciones de la evaluación andragógica, la cual debe ser ininterrumpida y diagnóstica. Las fallas, tan pronto como se detecten, deben ser corregidas. Por lo tanto, se contribuyó a definir un modelo de referencia de PC y un constructo andragógico que aborda un proceso de enseñanza y aprendizaje, que da como resultado la categorización del sujeto referente a sus habilidades de PC.

La *app* Inventor ha demostrado ser una herramienta adecuada para desarrollar la habilidad 4, ya que a partir de la existencia y frecuencia de bloques de instrucción, es posible analizar diferentes niveles de complejidad (Grover *et al.*, 2017), además de satisfacer las expectativas de los estudiantes y del mercado laboral.

Entonces, desarrollar habilidades de programación actualmente es importante. Según Pring *et al.* (2017), de cada 13 nuevas profesiones que surgirán, 10 exigirán el conocimiento de lenguajes de programación y obviamente de algoritmos, pues se está viviendo la edad del algoritmo (Rainie & Anderson, 2017).

Ya se están realizando nuevas aplicaciones del método en otros cursos binacionales, con el fin de validar los resultados descritos en este artículo. En nuevas aplicaciones del MRPC, el investigador puede aventurarse a ampliar el tiempo de aplicación e incluir un mayor número de preguntas para las pruebas, siempre cuidando de identificar correctamente a qué habilidad se refiere. Se cree que el marco propuesto en este trabajo puede ser utilizado

y/o adaptado como disciplina para ofrecer en los cursos de educación superior, en cualquier área, como soporte para el desarrollo del PC y el uso de tecnologías digitales. La educación de adultos puede —y quizás debería— ser un *continuum* educativo, coexistente con la vida y extendido a las dimensiones de una sociedad en constante cambio, un ejemplo bien visto en este año pandémico en el que trabajar desde casa marcó la diferencia para garantizar el empleo.

Referencias

- Ausubel, D. (2003). *Aquisição e retenção de conhecimentos: Uma perspectiva cognitiva. Tradução de Lígia Teopisto* (1a ed.). Plátano Edições Técnicas.
- Ausubel, D., Novak, J., & Hanesian, H. (1983). *Psicología Educativa Un punto de vista cognoscitivo* (2a ed.). Trillas.
- Brackmann, C. (2017). *Desenvolvimento do pensamento computacional através de atividades desplugadas na Educação Básica* [Tesis Doctoral, Universidade Federal do Rio Grande do Sul]. LUME Repositorio Digital. <https://lume.ufrgs.br/handle/10183/172208>
- Cordenonzi, W. (2020). *O desenvolvimento do pensamento computacional e as evidências da alfabetização em código em adultos* [Tesis Doctoral no publicada]. Universidade do Vale do Taquari - Univates.
- Conselho Nacional de Saúde. (2012, 12 de diciembre). Resolução N° 466. <https://conselho.saude.gov.br/resolucoes/2012/Reso466.pdf>
- Cordenonzi, W., Del Pino, J., Oliveira, E., & Guimarães Strohschoen, A. (2020). Alfabetização – uma evolução do conceito: alfabetização e letramento em código. *Texto Livre: Linguagem e Tecnologia*, 13(1), 137. <https://doi.org/10.17851/1983-3652.13.1.137-155>
- Cordenonzi, W., Ortiz, A., Meneses, L., Gau de Mello, A., Botasso, J., Mariani, V., Gomez, R., & Cross, V. (2014). *Mapeamento da Educação Binacional - Relatório Técnico I*. http://ecompartindo.santana.ifsul.edu.br/pluginfile.php/2328/mod_resource/content/3/relatorio_1_-_mapeamento_da_educacao_binacional.pdf
- Dagiené, V., & Futschek, G. (2008). Bebras International Contest on Informatics and Computer Literacy: Criteria for Good Tasks. En R. Mittermeir, & M. Sysło (Eds.). *Informatics Education - Supporting Computational Thinking* (pp. 19-30). Springer.
- Gil, A. C. (2002). *Como elaborar projetos de Pesquisa* (4a ed.). Atlas.
- Grover, S., Basu, S., Bienkowski, M., Eagle, M., Diana, N., Stamper, J., Grover, S., Basu, S., Bienkowski, M., Eagle, M., Diana, N., & Stam-Per, J. (2017). 4 A Framework for Using Hypothesis-Driven Approaches to Support Data-Driven Learning Analytics in Measuring Computational Thinking in Block-Based Programming Environments *ACM Transactions on Computing Education*, 17(3), 14. <https://doi.org/10.1145/3105910>
- Kalelioglu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, 4(3), 583. <https://www.sciencedirect.com/science/article/abs/pii/S0360131519301605>

- Knowles, M. (1981). *The modern practice of adult education: From pedagogy to andragogy* (2a ed.). Cambridge Book.
- Knowles, M., Holton, E., & Swanson, R. (2011). *Aprendizagem de Resultados* (2a ed.). Elsevier.
- Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558–569. <https://doi.org/10.1016/j.chb.2017.01.005>
- Lee, I., Grover, S., Martin, F., Pillai, S., Malyn-Smith, J., & Org, J. (2020). Computational Thinking from a Disciplinary Perspective: Integrating Computational Thinking in K-12 Science, Technology, Engineering, and Mathematics Education. *Journal of Science Education and Technology*, 29, 1–8. <https://doi.org/10.1007/s10956-019-09803-w>
- Moreno, J., Román, M., & Robles, G. (2018). On computational thinking as a universal skill: A review of the latest research on this ability. En *IEEE Global Engineering Education Conference (EDUCON)* (pp.1684–1689). <https://doi.org/10.1109/EDUCON.2018.8363437>
- Pring, B., Brown, R., Davis, E., Bahl, M., & Cook, M. (2017). *21 Jobs on the Future. A Guide to Getting – and Staying –Employed over the Next 10 Years*. Center for the Future of Work. <https://www.cognizant.com/whitepapers/21-jobs-of-the-future-a-guide-to-getting-and-staying-employed-over-the-next-10-years-codex3049.pdf>
- Raabe, A., Santana, A., Ellery, N., & Gonçalves, F. (2017). Um Instrumento para Diagnóstico do Pensamento Computacional. *Congresso Brasileiro de Informática Na Educação (CBIE)*, VI, (pp. 1172–1181). <https://doi.org/10.5753/cbie.wcbie.2017.1172>
- Rainie, L., & Anderson, J. (2017). *Code-Dependent: Pros and Cons of the Algorithm Age*. Pew Research Center. <http://www.pewinternet.org/2017/02/08/code-dependent-pros-and-cons-of-the-algorithm-age>
- Román, M. (2016). *Codigoalfabetización y Pensamiento Computacional en Educación Primaria y Secundaria: Validación de un Instrumento y Evaluación de Programas* [Tesis Doctoral, Universidad Nacional de Educación a Distancia]. E-spacio. <http://e-spacio.uned.es/fez/view/tesisuned.Educacion-Mroman>
- Román, M., Pérez, J., & Jiménez, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72, 678–691. <https://doi.org/10.1016/j.chb.2016.08.047>
- Santos, C. (2010). Andragogia: aprendendo a ensinar adultos. *VII Simpósio de Excelência Em Gestão e Tecnologia*. http://www.aedb.br/seget/arquivos/artigos10/402_ArtigoAndragogia.pdf
- Tikva, C., & Tambouris, E. (2021). A systematic mapping study on teaching and learning Computational Thinking through programming in higher education. *Thinking Skills and Creativity*, 41, 100849. <https://doi.org/10.1016/j.tsc.2021.100849>
- Triviños, A. (2015). *Introdução à pesquisa em Ciências Sociais - A pesquisa qualitativa em Educação*. Atlas.
- Wenger, E. (2002). *Cultivating communities of practice: a quick start-up guide for communities*

of practice. Harvard Business Review Press.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>

Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, 25(2), 7–14. <https://www.learntechlib.org/p/183466/>

Yadav, A., Burkhart, D., Moix, D., Snow, E., Bandaru, P., & Clayborn, L. (2015). *Sowing the seeds of assessment literacy in secondary computer science education: A landscape study*. Computer Science Teachers Association. <http://csta.acm.org/Research/sub/Projects/AssessmentStudy2015.html>

Agradecimiento

Al Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense, Brasil, por el apoyo financiero para realizar esta investigación doctoral.