

PENSAMENTO COMPUTACIONAL: Instrumentos Para Avaliar e Classificar a Alfabetização em Código

Walkiria Helena Cordenonzi¹
José Claudio Del Pino²

RESUMO

O objetivo deste artigo é apresentar um conjunto de instrumentos para avaliar o desenvolvimento do Pensamento Computacional em indivíduos adultos. Estes instrumentos foram ancorados na teoria da aprendizagem significativa (Ausubel) e na Andragogia (Knowles). A proposta consiste em apresentar um Modelo de Referência do Pensamento Computacional (MRPC) e um método de avaliação baseado em avaliações objetivas e subjetivas. Para sua validação foi desenvolvido um constructo andragógico. De acordo com o resultado da aplicação deste método, é possível classificar os participantes em Alfabetizado em Código, Pensador Computacional Desplugado ou nenhum dos anteriores. Esse método foi aplicado em cursos binacionais, nos quais brasileiros e uruguaios compartilham o mesmo espaço de aprendizagem. Por meio de dois estudos de caso e da análise quali-quantitativa dos resultados, obteve-se, do total da amostra, 66,6% dos sujeitos classificados como alfabetizados em código, e 14% como pensador computacional desplugado. Esse resultado evidencia a correteza dos instrumentos propostos e sua aplicabilidade.

Palavras-chave: Avaliação do pensamento computacional. Alfabetização em código. Constructo andragógico.

COMPUTATIONAL THINKING: INSTRUMENTS TO EVALUATE AND CLASSIFY CODE LITERACY

ABSTRACT

The aim of this article is to present a set of instruments to assess the development of Computational Thinking in adult individuals. These were anchored in theories of meaningful learning (Ausubel) and Andragogy (Knowles). The proposal consists of presenting an MRPC (Reference Model of Computational Thinking) and an evaluation method (based on objective and subjective evaluations). For its validation an andragogical construct was developed and as a result of applying this method, it is possible to classify the participants into Code Literate (ACod), Unplugged Computational Thinker (PCD) or none of the above. This method was applied in two binational courses - in which Brazilians and Uruguayans share the same learning space. Through case studies and qualitative and quantitative analysis of the results, 66.6% of the subjects were classified from the total sample as ACod and 14% as PCD. This shows the correctness of the proposed instruments and their applicability.

Keywords: Computational thinking assessment. Code literacy. Andragogical construct.

Recebido em: 25/9/2020

Aceito em: 11/2/2021

¹ Autora correspondente. Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense. R. Gonçalves Chaves, 3218 – Centro. CEP 96015-560. Pelotas/RS, Brasil. <http://lattes.cnpq.br/9583417199388924>. <https://orcid.org/0000-0001-9351-0920>. walkiriacordenonzi@ifsul.edu.br

² Univates – Universidade do Vale do Taquari. Av. Avelino Talini, 171 – Universitário. CEP 95914-014. Lajeado/RS, Brasil. <http://lattes.cnpq.br/2152799270731771>. <https://orcid.org/0000-0002-8321-9774>.

O processo de ensino e aprendizagem, desde muito tempo, vem sendo foco de pesquisas e experimentações. O estudo das Ciências começa a ser ensinada no Brasil, como componente curricular, a partir de 1930. Este estudo e a sua incorporação nos currículos da Ciência da Computação (CC), porém, ainda não foram totalmente evidenciados em nenhum dos níveis de ensino (CERF, 2012) neste país. As escolas precisam mudar no sentido de fazer a intersecção do conhecimento já instituído com o uso das tecnologias digitais. Esta convergência de temas também não é atual.

Wing, em 2006, apresenta uma nova designação – o pensamento computacional (PC) –, que pode ser entendido como o processo de resolver e identificar problemas, utilizando métodos e técnicas provenientes da ciência da computação. Ainda, segundo a autora, ao utilizar essas técnicas facilita-se a obtenção dos resultados, principalmente para encontrar soluções de problemas complexos. Na literatura encontram-se diferentes definições desta expressão; por isso, ainda hoje não há consenso em relação ao assunto. Posto isso, neste artigo segue-se a conceituação de Wing (2017), que entende o PC como o “processo de pensamento envolvido na formulação de um problema e em expressar sua(s) solução(ões) de tal forma que um computador – humano ou máquina – possa efetivamente ser executado” (p. 8, tradução nossa).

Esse tema, todavia, além de outros, como resolução de problemas, programação e raciocínio lógico, podem ser encontrados nos trabalhos de Papert (1980) na sua teoria sobre o construcionismo. É certo que o PC trata do desenvolvimento de habilidades cognitivas, entre elas a de resolver problemas utilizando métodos e técnicas computacionais, mas não se restringe somente a isso. Em outras palavras, desenvolver o PC na programação não significa apenas programar, mas, sim, desenvolver todo o processo para encontrar a solução de um problema.

É certo que, para programar, é necessário o desenvolvimento de certas habilidades que agregam a aprendizagem de algoritmos e, ainda, uma linguagem de programação.

O acesso fácil à *internet*, a popularização dos dispositivos móveis, principalmente os *smartphones*, e o uso de vários aplicativos disponíveis nas lojas virtuais, tornam os usuários consumidores de tecnologia. Estes mesmos atores, no entanto, podem ser capazes de também serem produtores de tecnologia; portanto deverão ser, pelo menos, alfabetizados em código. Rushkoff (2012, tradução nossa, texto digital), preocupado com o uso dos programas, sem entender o que estava acontecendo no ambiente digital, argumenta que a “alfabetização em código é um requisito mínimo para participar de um ambiente digital”. Em 2009, Resnick *et al.* denominavam esse conhecimento como “fluidez digital”. Seu trabalho centrava-se em ensinar crianças (a partir de 8 anos) e jovens a produzirem códigos por intermédio de uma ferramenta chamada *Scratch*³. Esses autores, porém, não descreveram nenhum método independente da linguagem de programação para avaliar e definir a alfabetização em código.

³ *Scratch* é uma linguagem gráfica de programação. Disponível em: <http://www.scratchbrasil.net.br/index.php/sobre-o-scratch.html>. Acesso em: 12 jun. 2018.

Incorporar o ensino de PC à educação formal já no início da educação, portanto, passa a ser bem importante para que os alunos comecem a dominar o PC e, consequentemente, a resolução de problemas.

Desse modo, une-se à preocupação com os processos de ensino e aprendizagem, para analisar como os sujeitos adultos desenvolvem seu pensamento computacional tanto para atender às demandas da sociedade quanto para lograr resolver problemas do cotidiano. Assim sendo, o ensino de PC associa-se à preocupação com as mudanças causadas pelo advento das tecnologias por parte de sujeito adulto, que estará ou já está no mercado de trabalho (RUIPÉREZ, 2017) e, concomitantemente, busca sua qualificação, incorporando novas habilidades, inclusive a do pensamento computacional. Dado esse cenário, traçou-se como objetivo deste artigo, que é resultado da tese de Doutorado da autora: compreender e analisar o desenvolvimento do PC nos sujeitos adultos com formação em escolas brasileiras e uruguaias. Para tanto, foi desenvolvido um Modelo de Referência de Pensamento Computacional (MRPC) e um método de avaliação para classificar os sujeitos quanto à alfabetização em código.

Este artigo está organizado da seguinte forma: na seção 2 apresenta-se o referencial teórico; na seção 3, descrevem-se os procedimentos metodológicos e os instrumentos de avaliação; na seção seguinte são detalhados e discutidos os resultados relativos à aplicação dos instrumentos; por fim, expõe-se as considerações finais e as referências bibliográficas.

REFERENCIAL TEÓRICO

Há diversos entendimentos e definições acerca do conceito de PC (do inglês, *Computational Thinking*), mas não há consenso, ainda, sobre o que é PC e o que ele implica. Partindo do exposto por Wing (2006), o PC é uma habilidade fundamental que pode ser utilizada por todas as pessoas do século 21, uma vez que se assemelha às habilidades de ler, escrever e calcular. O PC refere-se, portanto, às pessoas e como elas desenvolvem a habilidade de resolver problemas, independentemente dos recursos computacionais. Esta autora sugere que os cursos superiores ofereçam uma disciplina que aborde “formas de pensar como um cientista da computação”, não somente para iniciantes em atividades e tarefas de programação, mas para todos, independentemente do curso ou da idade do sujeito.

Mais recentemente, no Brasil, em 2018, com a proposta do Currículo de Referência em Tecnologia e Computação (CRTC) definida pelo Centro de Inovação para a Educação Brasileira (CIEB), percebe-se que já existe um movimento referente à inserção do PC na Educação Básica. Este currículo de referência, destinado à Educação Infantil e ao Ensino Fundamental, destaca que “cada uma das habilidades do Currículo de Referência em Tecnologia e Computação se relaciona com uma ou mais competências gerais e habilidades da Base Nacional Comum Curricular” (CRTC, 2018, p. 22) e define o PC como sendo o processo que compreende “sistematizar, representar, analisar e resolver problemas” (CRTC, 2018, p. 19). Seu principal objetivo é prover um suporte curricular para a comunidade educativa nos temas relacionados à tecnologia e à computação.

Além das definições já apresentadas, Brackmann (2017, p. 31) define PC como

[...] uma distinta capacidade criativa, crítica e estratégica humana de saber utilizar os fundamentos da Computação, nas mais diversas áreas do conhecimento, com a finalidade de identificar e resolver problemas, de maneira individual ou colaborativa, através de passos claros, de tal forma que uma pessoa ou uma máquina possam executá-los eficazmente.

Constata-se que esse autor define mais detalhadamente o conceito proposto por Wing, em 2006, ao trabalhar com o PC de forma desplugada, isto é, sem o uso do computador.

Em sua tese de doutorado, Román González (2016) enfatiza que a conceitualização de PC ainda não está estabelecida (BRENNAN; RESNICK, 2012; KORKMAZ; ÇAKIR; ÖZDEN, 2017; KOTSOPOULOS *et al.*, 2017; ROMÁN GONZÁLEZ; PÉREZ GONZÁLEZ; JIMÉNEZ FERNÁNDEZ, 2017), isto é, ainda não há um consenso de como incorporar este tipo de pensamento ao currículo escolar, nem como avaliá-lo. Analisando as definições de PC propostas por Brennan e Resnick (2012), CRTIC (2018), Ribeiro, Foss e Cavalheiro (2017) e Wing (2006, 2017), as habilidades de maior concordância são as de “resolução de problemas”, “abstração” e “algoritmos”.

É possível afirmar, todavia, que a abstração pode ser considerada como uma das mais destacadas habilidades (RIBEIRO; FOSS; CAVALHEIRO, 2017), sendo indiscutível sua importância para o desenvolvimento do PC (GROVER; PEA, 2013).

Para reforçar o entendimento do conceito de PC e suas implicações sobre as habilidades mais importantes, Avila *et al.* (2017) analisaram 58 artigos e chegaram à mesma conclusão: as habilidades mais trabalhadas foram o pensamento algorítmico, a resolução de problemas e a abstração. Já Kalelioglu, Gülbahar e Kukul (2016) analisaram 125 artigos, recuperados de fontes digitais e bases de dados, examinando a definição, o escopo e a base teórica do pensamento computacional (PC), e concluíram que os conceitos mais utilizados para descrever o significado de PC foram os seguintes: 22% como resolução de problemas e 13% como um processo de abstração. Já o pensamento algorítmico aparece em sexto lugar, com 6% do total das definições.

Numa pesquisa realizada no Banco de Teses e Dissertações da Capes (BTD), com a palavra-chave “pensamento computacional” (em 25-4-2020), resultaram 15 teses de Doutorado, das quais somente 5 trabalharam com adultos (neste contexto entendido como alunos de cursos pós-médio e superior). Destas, duas na área de Ciência da Computação, uma em Informática na Educação, uma defendida na Informática, e outra na Psicologia. Esta tese, que resultou neste artigo, é a primeira na área de Ensino e a terceira que analisa o desenvolvimento do PC por meio de uma abordagem qualiquantitativa em adultos.

Dentre algumas experiências com a avaliação do PC em adultos, pode-se citar a pesquisa de Korkmaz, Çakir e Özden (2017), que definiram uma escala que determina os níveis de Habilidade de Pensamento Computacional (*computational thinking skills*) e a aplicaram em alunos da Amasya University, na Turquia. Esta escala está dividida em 29 itens, distribuídos em fatores. Este trabalho é uma das poucas referências sobre APC em estudantes que já concluíram a Educação Básica. Ainda assim, os autores não descreveram como o método foi aplicado e não apresentam dados de quanto os estudantes desenvolveram as habilidades.

No trabalho de Avila *et al.* (2017), os autores apresentam uma revisão sistemática da literatura, entre os anos de 2011 e 2016, sobre avaliação do PC. Dos 58 artigos considerados, 65% foram classificados em intervenção com avaliação própria e intervenção com avaliação própria fundamentada em métodos existentes. Não consideram em sua pesquisa, porém, o nível de ensino nos quais os testes foram aplicados, tampouco diferenciam a avaliação realizada em relação ao uso de ferramentas.

Moreno León, Román González e Robles (2018) salientam que os métodos mais utilizados pela comunidade educativa são *CT-Test*, *Bebras* e *Dr. Scratch*, que são empregados para avaliar o PC sob diferentes pontos de vistas, bem como de forma conjunta, sendo compatíveis e complementares.

Na sua tese, Araújo (2019) propôs-se a investigar estratégias e instrumentos para quantificar o PC sem o uso de práticas de programação. Como resultado, desenvolveu um modelo baseado em estudos empíricos para quantificar o PC como uma habilidade cognitiva. Este modelo é dividido em 4 competências e 12 habilidades, sendo os sujeitos investigados alunos matriculados em duas universidades.

Muitas outras pesquisas sobre avaliação do PC foram encontradas (KATAI; TOTH, 2010; KORDAKI, 2010; MORENO, 2012; MORENO LEÓN; ROBLES; ROMÁN GONZÁLEZ, 2015; RODRIGUES *et al.*, 2015; SEITER; FOREMAN, 2013), mas elas não estão descritas, pois os níveis de ensino abordados não são semelhantes ao do escopo desta investigação.

Brennan e Resnick (2012) propuseram um *framework* de três dimensões: conceito, prática e perspectiva computacionais. Os autores sugerem conceitos e práticas a serem desenvolvidos, porém muitos conceitos estão definidos, mas, na prática, utilizam o *Scratch* com o nível K-12. Esta pesquisa utilizou parte deste *framework* para a definição do seu Modelo de Referência do PC (MRPC).

Em síntese, o campo de avaliação do PC ainda está carente de pesquisas. Foi possível constatar que boa parte das metodologias propostas é do tipo qualitativo (RAABE *et al.*, 2017) e que há poucas ferramentas para avaliação que produzem *feedback* imediato, as quais estão restritas ao ambiente de desenvolvimento utilizado. Além disso, o desenvolvimento de metodologias de aplicação e de avaliação do PC para crianças e jovens (que cursam o Ensino Fundamental) contam com maior referencial teórico e prático se comparado ao nível universitário ou adultos, conforme afirmam Tang *et al.* (2020). Na pesquisa realizada por esses autores, em 77 artigos anteriores a agosto de 2019, apenas 15% estão focadas no Ensino Superior. Assim, concluem que ainda há poucas publicações na área de PC, sugerindo, assim, a integração de várias ferramentas para melhorar a avaliação da aprendizagem com foco no PC.

A partir do entendimento de que os processos de ensino e aprendizagem (são dois processos que podem ser independentes; por exemplo, se ensina, mas não se aprende) de adultos é diferente dos de crianças, na seção seguinte comentam-se alguns conceitos importantes sobre as teorias de aprendizagem que embasaram este trabalho.

TEORIA DE APRENDIZAGEM

O objetivo desta seção é apresentar alguns aspectos relevantes utilizados para embasar esta pesquisa, a saber, a Aprendizagem Significativa e a Andragogia.

Ausubel (1968) propõe uma teoria cognitivista que se preocupa com os mecanismos internos da mente, cujo principal conceito é a aprendizagem significativa. Ele entende que a estrutura do conhecimento está organizada de forma hierárquica no sujeito, sendo um conjunto de conceitos, de ideias ou proposições que fazem parte de uma área de conhecimento que estão interacionados.

O sujeito (ou aprendiz) aprende quando ancora um novo conhecimento em sua estrutura cognitiva. O fator mais importante na aprendizagem é o conhecimento prévio do aluno. Ausubel resume sua teoria na seguinte frase: “o fator mais importante que influencia a aprendizagem é o que o aluno já sabe. Descubra isso e ensine-o de acordo” (AUSUBEL; NOVAK; HANESIAN, 1983, p. 1). Esse processo ocorre por meio de uma nova informação que interage de forma não literal e não arbitrária. Não literal significa que a relação entre o material a ser aprendido e a estrutura cognitiva não deve ser ao pé da letra (ou seja, a relação não se altera se símbolos diferentes, mas equivalentes, forem usados). Já por não arbitrária entende-se que o significado está na relação do novo item com a estrutura cognitiva. Esse significado não deve ser ao acaso, mas feito de modo intencional, para permitir a ancoragem (conexão) com os subsunçores específicos. Os subsunçores são as estruturas cognitivas que o aprendiz já possui e que permitem a ligação dos novos conhecimentos. Em decorrência da ancoragem, os subsunçores podem ser modificados; portanto, pode ser entendido como um ciclo: o aprendiz recebe uma informação, que é ancorada em um ou mais subsunçores, que são modificados e outra vez recebem novas informações e reinicia-se, assim, o processo de cognição (SILVA; DEL PINO, 2019), que pode ser definido como “dinâmico e o conhecimento vai sendo construído” (MOREIRA, 2010, p. 18).

Em 1981, Knowles (1981, p. 43) definiu a andragogia “como a arte e a ciência de ajudar os adultos a aprender, em contraste com a pedagogia como a arte e a ciência de ensinar as crianças”. À medida que cresce, o indivíduo desenvolve suas habilidades, esperando uma aplicação prática do conhecimento, resolvendo problemas reais ligados ao seu cotidiano ou com possíveis melhorias em sua vida (KNOWLES; HOLTON; SWAMSON, 2011). Passa a ser autodirigido, utilizando seus conhecimentos e sua própria capacidade de aprender. Malcolm Knowles propôs um conjunto de princípios sobre a aprendizagem de adultos e passa a ser considerado o pai da andragogia nos Estados Unidos (KNOWLES; HOLTON; SWAMSON, 2011), defendendo que “não se pode ensinar um adulto, mas, sim, ajudá-lo a aprender” (KNOWLES; HOLTON; SWAMSON, 2011, p. 13).

O ponto essencial da Andragogia repousa, portanto, num conjunto de princípios de aprendizagem de adultos, aplicáveis a qualquer situação. Nesta fase, o importante é experienciar e vivenciar os momentos para que se tornem significativos, ou seja, aprender fazendo. A seguir a descrição dos procedimentos metodológicos.

PROCEDIMENTOS METODOLÓGICOS

A fim de retomar os princípios essenciais desta pesquisa, cujo objetivo é compreender e analisar o desenvolvimento do PC nos sujeitos adultos, com formação em escolas brasileiras e uruguaias, inicialmente faz-se necessário entender que os sujeitos desta pesquisa são os alunos que atualmente estão matriculados num curso binacional pós-médio, com a finalidade de englobar os cursos técnicos no nível subsequente e os cursos superiores. Já a questão da formação em escola brasileira ou uruguaia se dá pela nacionalidade do sujeito. Esta diferenciação está calcada nos cursos binacionais, que

surgiram de um acordo binacional entre os governos brasileiro e uruguaio, representado pelo Consejo de Educación Técnico Profesional – Universidad del Trabajo del Uruguay (CETP-UTU) e pelo Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense (IFSUL) (CORDENONZI, 2020), respectivamente. O que também caracteriza esses cursos é a dupla diplomação. Inegavelmente, este contexto de trabalho é singular, a começar pelo uso de dois idiomas (maternos ou L1) em sala de aula.

Este trabalho foi realizado seguindo o método indutivo (GIL, 2008, p. 10). Quanto à abordagem do problema, esta pesquisa é quali-quantitativa. Importa salientar que os pesquisadores qualitativos não estão preocupados somente com os resultados esperados e/ou encontrados ou somente com o produto, mas também com todo o processo e com a atribuição de significados. Já as pesquisas quantitativas estão centradas na objetividade, em que a análise dos dados está centralizada nos números e nos seus significados. Gatti (2004, p. 13) confirma que estes significados “podem ser muito úteis na compreensão de diversos problemas educacionais”. Com a combinação de ambas as abordagens, consegue-se recolher mais dados e informações “muito úteis na compreensão de diversos problemas educacionais” (GATTI, 2004, p. 13). O método descritivo baseia-se nos objetivos. Corroborada por Triviños (2015) e Gil (2008), a maioria das pesquisas no campo da educação são desenvolvidas de forma descritiva. Ainda, no campo dos procedimentos técnicos, foi utilizado o estudo de caso (GIL, 2008). Na primeira etapa da pesquisa foram definidos os instrumentos para analisar o desenvolvimento do PC e sua avaliação nos sujeitos. Na segunda etapa foram realizados os estudos de caso e, por fim, os dados foram analisados para a obtenção dos resultados. Todo o processo está descrito na sequência do texto.

Modelo de referência para o desenvolvimento do PC – MRPC

Na literatura há muitas divergências no que se refere à definição e ao entendimento de habilidades e competências (FROTA, 2017). No contexto desta pesquisa o entendimento é que as habilidades estão associadas ao saber-fazer, ou seja, pode ser uma ação física ou mental que aponta para uma capacidade adquirida (MORETTO, 1999). Já as competências são interpretadas como a faculdade de mobilizar o saber-fazer a fim de “solucionar com pertinência e eficácia uma série de situações” (PERRENOUD, 1999, p. 30), como também como um conjunto de habilidades desenvolvidas uniformemente, conforme bem-esclarece Garcia (2005, p. 5):

[...] uma pessoa, por exemplo, que tenha uma boa expressão verbal (considerando que isso seja uma habilidade) pode-se utilizar dela para ser um bom professor, um radialista, um advogado, ou mesmo um demagogo. Em cada caso, essa habilidade estará compondo competências diferentes.

Nesta pesquisa sugere-se as seguintes habilidades (mínimas) que o sujeito deverá apresentar para desenvolver o PC: compreensão, abstração, resolução de problemas, resolução algorítmica e avaliação, como se explica na sequência.

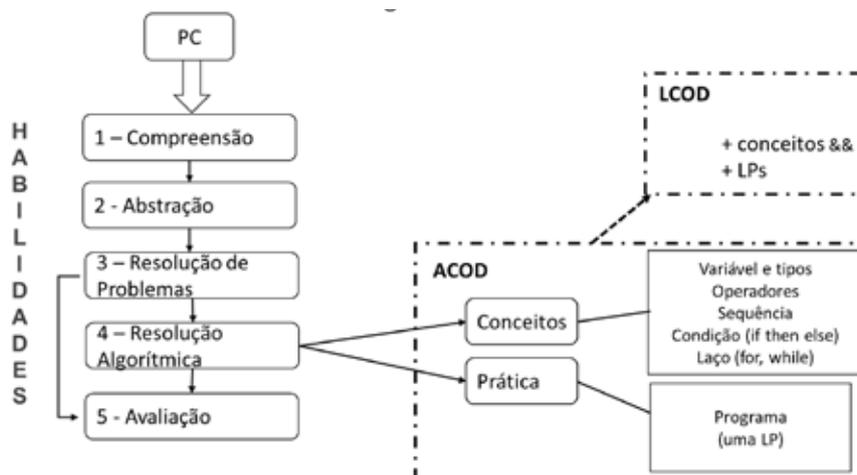
1. *Compreensão*: defende-se que este é o primeiro passo para a resolução de um problema. Segundo Opmanis, Dagiene e Truu (2006, p. 5, tradução nossa), “um aspecto importante de qualquer tipo de manipulação de informações é a capacidade de

compreender o significado da informação (em oposição ao processamento meramente mecânico)”. Inicialmente, o sujeito precisa entender a proposição do problema e o contexto para estar apto a significar e a aplicar, estabelecendo uma organização hierárquica entre a compreensão e a habilidade de abstração. A habilidade de compreensão passa por uma aprendizagem significativa, na qual o sujeito constrói seu conhecimento e os seus processos cognitivos levam à resolução bem-sucedida do problema. No primeiro passo, o sujeito lê e compreende o que foi lido, podendo utilizar seus conhecimentos prévios (subsunçores), pois, para Ausubel (2003, p. 7), é o “fator determinante do processo de aprendizagem”. Além disso, é necessário que o sujeito encontre a relação entre os conhecimentos prévios e os utilize no passo seguinte, no qual deverá obter a solução do problema. Para isso, necessita de outras habilidades (apresentadas a seguir, por exemplo, abstração e resolução de problemas). Isto posto, a “compreensão”, uma habilidade proposta nesta pesquisa a fim de apoiar os constructos definidos, é considerada de grande importância andragógica, pela qual se pode perceber o entendimento do sujeito no que se refere à resolução de um problema, que é um dos fatores críticos para o desenvolvimento do PC.

2. *Abstração*: em 2006, Wing (2006) definiu a “abstração” como o processo que decide quais detalhes deverão ser destacados e quais descartados, e complementa afirmando que é o processo de mais alto nível quando se pensa em PC (WING, 2017). Neste ponto, há uma discordância, pois se acredita que, para o sujeito executar o processo de abstração, precisa necessariamente ter compreendido o problema; desta forma, a abstração fica como o próximo passo.
3. *Resolução de problemas*: centra-se na capacidade do sujeito de encontrar uma ou várias soluções, independente do formato da sua resposta. Da mesma forma, e justificando sua importância, Ausubel, Novak e Hanesian (1983, p. 448, tradução nossa) descreveram que “a capacidade de resolver problemas constitui a meta primária da educação”.
4. *Resolução algorítmica*: deve-se definir o que é um algoritmo e o que é um programa. Os algoritmos são uma sequência de passos lógicos, descritos em qualquer linguagem, geralmente em linguagem natural (por exemplo, língua portuguesa). Já os programas (ou *softwares*) são a “tradução” dos algoritmos para uma linguagem de programação. Essa habilidade, portanto, está na prática do sujeito a partir da habilidade anterior, na qual se encontrou a solução do problema, traduzindo-o para um algoritmo ou *software*.
5. *Avaliação*: é a capacidade do sujeito, após apresentar a resolução do problema por meio de algoritmo e/ou programa, de testar e avaliar a sua corretude. Em outras palavras, é entender o que foi instituído, ou seja, é a aptidão de analisar um código-fonte ou executável quanto às suas saídas. Entende-se, também, como avaliação ou teste, uma proposta de solução de um problema, independente do seu formato.

A partir de uma organização lógica das habilidades, propõe-se um modelo de referência para o desenvolvimento do PC (MRPC), que está representado de forma gráfica na Figura 1.

Figura 1 – Modelo MRPC



Fonte: Os autores (2019).

Seguindo a Figura 1, pode-se perceber a existência de uma dependência lógica, sugerida e não obrigatória, para desenvolver o PC. A partir da compreensão apropriada, a abstração passa a ser uma habilidade que, naturalmente, será mobilizada. Ou seja, compreendendo a situação, espera-se que o sujeito identifique os pontos importantes e descarte os que não têm utilidade. Após esta organização, o sujeito passa a focar seus esforços na busca de uma (ou mais) solução(ões) adequada(s). Neste ponto, não é importante saber quais métodos ou técnicas foram utilizados pelo sujeito para encontrar o resultado ou suas representações. Quando encontra o resultado (habilidade 3) e este pode ser avaliado ou testado (habilidade 5), entende-se que o processo de desenvolvimento do PC aconteceu, porém de uma forma desplugada. Em outras palavras, o formato da solução não foi por intermédio de um algoritmo. Caso o caminho escolhido pelo sujeito tenha sido por meio de uma solução algorítmica (Habilidade 4), definido como o “entendimento, aplicação, avaliação e produção” de um conjunto de instruções realizadas passo a passo para resolver uma tarefa (BROWN, 2015, texto digital, tradução nossa), logo passa-se a necessitar do domínio de determinados conceitos e práticas, ambos baseados no *framework* de Brennan e Resnick (2012).

Os conceitos que o programador deverá conhecer são estes: variáveis, constante, operadores (lógicos, relacionais e matemáticos) e as estruturas condicional e de repetição. Não basta, contudo, apenas conhecer esses conceitos de programação, mas também deve saber aplicá-los (momento da prática). Como resultado, obtém-se um programa. Em outras palavras, significa que, a partir da proposta de solução de forma algorítmica utilizando esses conceitos, é preciso que este algoritmo seja traduzido para uma linguagem de programação, tornando-se, então, um *software*. A partir do programa, ou do algoritmo, devem ser feitos os testes para verificar se o resultado está correto; portanto, a habilidade de avaliação também é necessária, conforme pode ser visualizado no MRPC.

Ao apresentar a habilidade 4 e baseando-se nos instrumentos avaliativos propostos, pode-se conferir ao sujeito o *status* de ACod, ou seja, está alfabetizado em código. No âmbito desta pesquisa, entende-se que:

- Um indivíduo *Alfabetizado em Código* (ACod) é aquele capaz de ler, interpretar e escrever um código-fonte e gerar um código executável (programa).
- Um indivíduo *Letrado em Código* (LCod) é um ACod extrapolando suas habilidades e competências para codificar em diferentes linguagens de programação (CORDENONZI *et al.*, 2020).

Numa primeira situação, considerando que o sujeito/aprendiz nunca teve contato com a habilidade 4 e com programação, os processos de ensino e aprendizagem devem ocorrer de forma a tornar significativos os conceitos e relacioná-los com a prática, principalmente em se tratando de aprendizagem significativa de adultos. Quando o sujeito extrapola esses conhecimentos, ou seja, é capaz de efetuar a sua prática em outras linguagens de programação, pode ser considerado letrado em código (LCod); este não será abordado neste trabalho.

A partir do entendimento deste modelo, tendo como base a aprendizagem significativa, a andragogia e o contexto dos alunos de cursos binacionais, foi desenvolvida uma proposta de curso para o qual foram planejados materiais instrucionais, organizados em documentos (tutoriais, códigos-fonte, entre outros) para serem utilizados nas aulas, e instrumentos para avaliação, além de um método de avaliação, conforme apresentado no texto que segue.

Constructo Andragógico

Foi proposto o curso “Eu Programa 1.0!”, com duração de 20 horas. A recomendação é que, a cada aula, um ou mais conceitos sejam apresentados de forma rápida ao aprendiz, quer dizer, não mais do que 10 minutos de apresentação. Segue-se a sua aplicação, em suma, a prática (programação). Knowles (1981) observou que as informações lembradas são as recebidas nos primeiros 15 minutos de aula.

Para executar a prática com os alunos foi escolhido o *software App Inventor*. A escolha dessa ferramenta deu-se porque, a partir de uma interface simples e intuitiva, o usuário é capaz de desenvolver seus aplicativos em pouco tempo. Em outras palavras, já nos primeiros encontros o aluno consegue desenvolver um aplicativo simples. No desenvolvimento de cada encontro, basicamente, deve-se observar o seguinte roteiro: apresentação e/ou retomada dos conceitos; construção do App e teste, que podem acontecer de forma paralela. Após construir o App em aula, sempre é sugerido ao aluno um desafio, a fim de incentivar a aprendizagem por descoberta, sugerida por Ausubel (2003). Este desafio consiste em propor uma melhoria ou alteração no aplicativo, a fim de verificar se houve aprendizagem por parte do discente.

Em síntese, quanto à escolha dos temas dos aplicativos, o modelo andragógico de Knowles foi um dos principais referenciais, fomentando no aprendiz a necessidade do saber. Para cada App proposto, o professor, ao terminar seu encontro, disponibiliza uma possível solução para o problema. Quando o aluno termina o seu App, envia-o para o professor, que faz a correção e retorna com um *feedback*, porque “sabemos que a andragogia deve ser mais personalizada para se adequar à singularidade entre os adultos” (KNOWLES; HOLTON; SWAMSON, 2011, p. 146), assim o foco da aprendizagem está no aluno e cabe ao professor/tutor acompanhar esse processo.

No curso, o aprendiz desenvolve seis Apps, um projeto final e é convidado a responder a vários testes, a fim de formar a sua avaliação, conforme descrito a seguir.

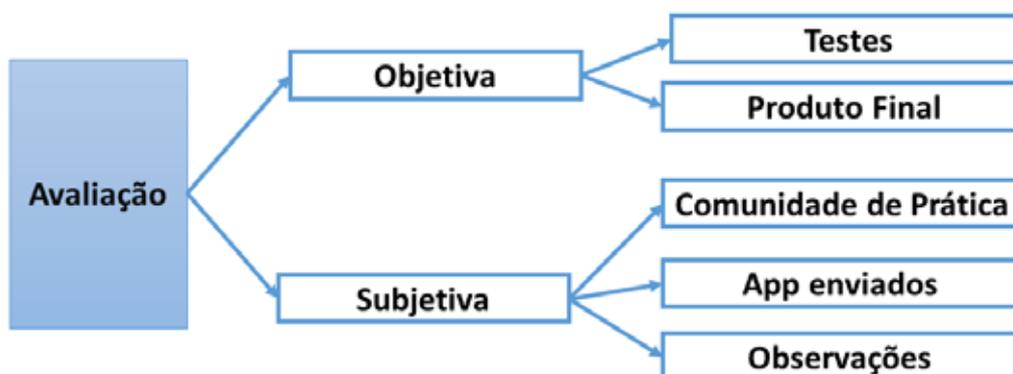
Instrumentos de avaliação

Como contribuição desta pesquisa, são propostos alguns instrumentos avaliativos com a finalidade de responder a essa questão.

Como avaliar não é um processo simples, propõem-se vários instrumentos, agrupados em dois conjuntos: instrumentos objetivos e subjetivos, conforme podem ser visualizados de forma gráfica na Figura 2.

Os (instrumentos) objetivos são compostos por testes (pré, pós e intermediários) e pelo produto resultante do projeto final, com seus respectivos indicadores. Os instrumentos subjetivos são: observação, comunidade de prática (CoP) e aplicativos desenvolvidos durante o curso. Essa divisão serve para o entendimento da proposta avaliativa de forma estática.

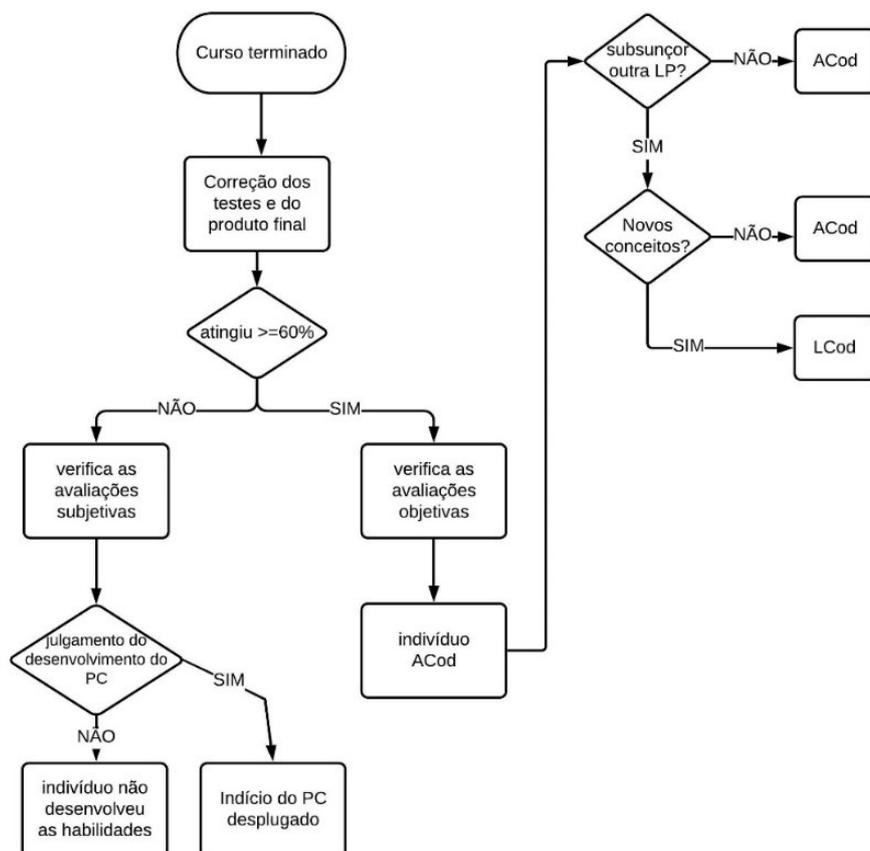
Figura 2 – Tipos de instrumentos avaliativos



Fonte: Os autores (2019).

Na Figura 3 apresenta-se o processo de avaliação por meio de um fluxograma.

Figura 3 – Processo de avaliação para o MRPC



Fonte: Os autores (2020).

Após todos os testes realizados e o projeto final (App) entregue e devidamente avaliado, faz-se a média dos valores encontrados, devendo o resultado atingir o valor mínimo de 60% de acertos. Com esses valores conhecidos, pode-se sugerir o *status* do indivíduo quanto à alfabetização em código. Já na avaliação subjetiva, o professor deve avaliar, de forma individual, cada aluno para poder determinar se o discente desenvolveu todas as habilidades propostas no MRPC, junto com as observações realizadas durante todo o curso e na CoP. O tutor poderá sugerir que o aluno desenvolveu o PC, porém de forma desplugada (PCD), ou seja, a Habilidade 4 não foi desenvolvida; em outras palavras, não resolveu os problemas de forma algorítmica ou mediante programas de computador.

Para avaliar o resultado do *App* proposto no projeto final do curso, buscou-se a norma ISO/IEC 25010 (2011)⁴ (*System and Software Quality Models*), que apresenta a definição de oito características de qualidade de *software* e suas subcaracterísticas, que são apresentadas para determinar o escopo de cada medida. Foram utilizadas as características: Funcionalidade (com as subcaracterísticas completude, correção e adequação); Usabilidade (reconhecimento, aprendizagem, operabilidade, proteção contra erros do usuário e estética); e Manutenibilidade (testabilidade).

⁴ Disponível em: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>. Acesso em: 2 ago. 2018.

Avaliação Subjetiva

São propostas três formas de avaliar o processo de desenvolvimento do PC: observação direta, acompanhamento dos aplicativos desenvolvidos durante o curso e a interação do aluno na CoP.

A observação direta é uma evidência que deve ser realizada pelo professor durante as aulas, o que significa observar os alunos em tempo real, refletir sobre e avaliar o posicionamento do estudante de acordo com o contexto e o conteúdo que estiver sendo trabalhado. Quanto ao segundo tipo, análise do código-fonte, o professor poderá acompanhar a produção do aluno com relação ao *software* desenvolvido, verificando a corretude e a completude. Na sequência, fornecer um *feedback* adequado, pois, indubitavelmente, ao analisar o código-fonte, pode-se perceber a presença (ou não) das habilidades referentes àquele App. No terceiro tipo, agregando à observação e à análise de desenvolvimento de cada aluno, cabe ao professor incentivar o uso da CoP, a fim de que o grupo compartilhe suas experiências. Nesse sentido, deve o professor observar as interações dos alunos para compor seu perfil em relação ao desenvolvimento do seu PC, por meio das habilidades sugeridas no MRPC. A partir desses instrumentos, nos quais a atuação do professor é importante e fundamental, cabe-lhe o julgamento referente ao conhecimento e ao desenvolvimento do PC do aluno. O parecer versará sobre se há indícios de melhorias no PC do aluno ou se não é possível evidenciar as habilidades sugeridas no modelo.

De um lado, ancorado em Ausubel (2003, 1983) e Knowles, Holton e Swanson (2011) e, de outro, no entendimento e na avaliação do PC, propôs-se estes constructos andragógicos, que tratam tanto de desenvolvimento de habilidades para resolver problemas na forma de programas ou não quanto para avaliar este processo. O resultado deverá mostrar indícios das habilidades propostas, com a finalidade de afirmar que o sujeito tornou-se um ACod, ou desenvolveu as habilidades, exceto a de programação (PCD – pensador computacional desplugado).

RESULTADOS E DISCUSSÕES

Os sujeitos envolvidos nesta pesquisa foram escolhidos pela particularidade de serem “alunos binacionais”, denominação derivada da matrícula deles em cursos binacionais. Estes cursos são o resultado de muitas negociações e relações bilaterais entre o governo brasileiro e o uruguaio para as cidades gêmeas de fronteira. Entre elas, Sant'Ana do Livramento (Brasil) e Rivera (Uruguai).

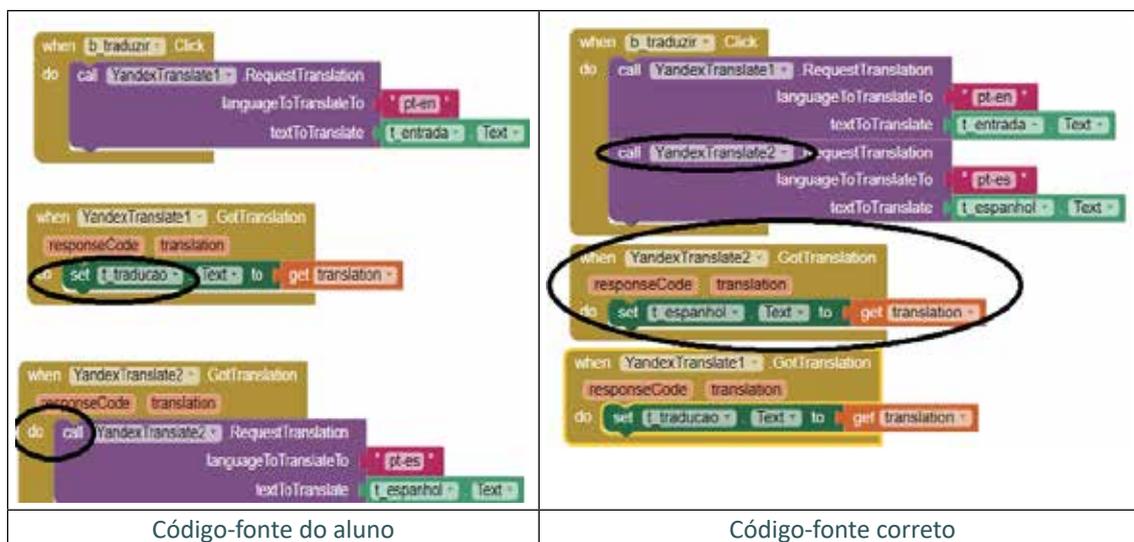
Foram realizados dois estudos de caso: o primeiro num centro educativo no Uruguai, e o outro numa escola no Brasil. O primeiro estudo (EC1) contava com 10 alunos matriculados, com uma média de idade de 23 anos, sendo 40% homens, 60% mulheres e 80% do total de sujeitos de nacionalidade uruguaia. Nenhum deles declarou ter algum conhecimento referente à programação. Deste total, dois alunos desistiram do curso e dois não entregaram o projeto final; portanto foram desconsiderados nos resultados finais.

No segundo estudo de caso (EC2) realizado, participaram 21 sujeitos. Destes, 47% são brasileiros e 53%, uruguaio; a média de idade é de 22 anos; 81% são homens e 19%, mulheres.

Para ambos os estudos de caso foram feitos vários testes, inclusive, pré e pós-teste, sendo construídos o contrato de aprendizagem e a definição do projeto. Quanto à avaliação subjetiva, a comunidade de prática foi instituída por meio do Fórum do Ambiente Virtual de Aprendizagem (Ava) e no *WhatsApp*. Os códigos-fonte dos Apps enviados pelos alunos foram analisados e, em seguida, enviados os *feedbacks* (compostos por anotações de texto e sinalizadores no código-fonte), totalizando mais de 150 códigos analisados.

Na Figura 4 está demonstrado um exemplo da análise do código-fonte do App Tradutor do aluno A18 do segundo estudo de caso.

Figura 4 – Código-fonte do App Tradutor do A18 e *feedback*



Fonte: Os autores (2020).

Na análise dos códigos-fonte apresentados na Figura 4, constata-se que o aluno não ancorou corretamente o uso dos comandos e sua sequência lógica, pois a chamada ao componente Tradutor (*YandexTranslate*) deve ser colocada quando o botão de tradução for acionado (*b_traduzir*), e não após ter sido realizada a tradução.

O resumo sobre a quantidade de sujeitos envolvidos nesta pesquisa, organizados por estudo de caso, é demonstrado no Gráfico 1.

Gráfico 1 – Sujeitos X Nacionalidade



Fonte: Os autores (2020).

A partir do exposto, pode-se inferir que o total de sujeitos envolvidos (entende-se como a soma dos participantes do EC1 e do EC2) foi de 27 alunos, sendo 12 brasileiros e 15 (ou seja, 55,55%) de nacionalidade uruguaia. Além disso, no primeiro estudo trabalhou-se com seis sujeitos e, no segundo, foram envolvidos 21 alunos.

Já no Gráfico 2 pode-se perceber a distribuição dos sujeitos catalogados por sua nacionalidade e classificação quanto ao desenvolvimento do PC.

Gráfico 2 – Alfabetização x Nacionalidade



Fonte: Os autores (2020).

Infer-se, a partir do Gráfico anterior, que o número de sujeitos classificados como PCD é igual em ambas as nacionalidades. Do total da amostra estudada, 11,1% são brasileiros e 7,4% são uruguaiois. Ou seja, para 18,5% do conjunto de sujeitos não foi possível encontrar evidências de desenvolvimento de habilidades do PC.

Ao analisar a coluna dos sujeitos ACod (Gráfico 2), verifica-se que do total dos sujeitos avaliados nos estudos de caso (N=27), 66,66% foram classificados como alfabetizados em código, sendo 25,9% brasileiros e os demais uruguaiois (40,7%). Seguindo esta classificação, dos 12 sujeitos brasileiros envolvidos nos estudos de caso, 58,33% foram classificados como Acod; em relação aos alunos uruguaiois, o índice sobe para 73,33% (no caso, 15).

Tabela 1 – Resumo dos dados descritivos de EC1 e EC2

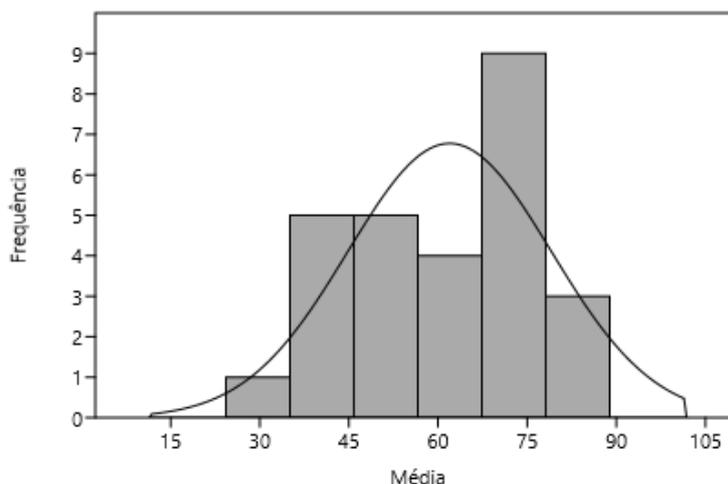
	Média Final	Desvio Padrão (DP)	Coefficiente de Variação – (CV)	Min	Max	Mediana	Moda
EC1	59,43	13,88	23%	38,67	72,33	66,78	66,78
EC2	62,66	18,18	29%	24,31	88,89	71,53	77,78
EC1+EC2	61,9	17,12	28%	24,31	88,89	66,78	66,78

Fonte: Os autores (2020).

A partir do exame dos dados da Tabela anterior, pode-se inferir que o valor final da avaliação objetiva (coluna Média Final) ficou acima dos 60% mínimos sugeridos pelo modelo proposto. Ou seja, a proposta do MRPC e sua aplicação no EC1 e EC2 podem ser consideradas corretas. Quanto às medidas de dispersão, o desvio padrão (DP) e o coe-

ficiente de variação (CV) indicam uma distribuição de dados não homogênea, classificada como regular em relação à média. Ainda mostram que a amostra de EC1 foi menos dispersa no segundo estudo de caso. Os valores que aparecem na coluna Mediana são sempre maiores que 60. Pode-se concluir, então, que o constructo andragógico empregado com os sujeitos obteve um bom resultado, pois mais da metade dos alunos atingiu o valor mínimo de 60%. Já a Moda, última coluna da Tabela 1, apresenta um valor final igual à Mediana. A mesma conclusão acerca do uso adequado do constructo andragógico, portanto, é novamente corroborada. O Gráfico 3 ilustra os dados das médias de todos os sujeitos envolvidos na pesquisa.

Gráfico 3 – Histograma da Média de todos os sujeitos



Fonte: Os autores (2020).

Quadro 1 – Dados da Amostra

<i>Amostra Total</i>	
Média	61,938
Mediana	66,778
Moda	77,778
Desvio Padrão	17,119
Variância da Amostra	293,055
Coefficiente de assimetria	-0,424
Mínimo	24,306
Máximo	88,889
Nível de confiança (95,0%)	6,772

Fonte: Os autores (2020).

A partir do histograma das médias, apresentado no Gráfico 3, pode-se inferir que a distribuição dos dados ocorreu de forma assimétrica à esquerda⁵, corroborado o coeficiente de assimetria de -0,424 (Quadro 1). Isso quer dizer que a maior concentração de dados está nos valores mais altos (direita).

⁵ O coeficiente de assimetria negativo (especialmente inferior a -1) indica que a cauda do gráfico é mais longa para a esquerda (BARBETTA, 2002).

Com base nos resultados obtidos, pode-se confirmar a utilização dos testes paramétricos, para os quais é exigido que as amostras dos dados tenham uma distribuição normal, principalmente se tiverem uma dimensão (total da amostra) inferior a 30 (MORAES, 2016). Utilizou-se, nesta pesquisa, o teste *Shapiro-Wilk* para testar a normalidade da amostra, que apresenta melhor desempenho para discriminar a distribuição de amostra normal (RAZALI; WAH, 2011), obtendo como resultado que a amostra provém de uma distribuição normal.

Segundo Campos (2000), o teste de *Teste t (Student)* é o mais comumente utilizado na prática para duas amostras.

Considerando as seguintes hipóteses: **H0**: $\mu_{EC1} = \mu_{EC2}$; **H1**: $\mu_{EC1} \neq \mu_{EC2}$, observando que o valor de $t_{\text{estatístico}}$ é igual a -0,401 e o valor crítico de t (bicaudal) é de 2,06, o primeiro valor se encontra no intervalo de -2,06 e +2,06, indicando uma distribuição normal. Usa-se o valor de t bicaudal ou bilateral, pois, na hipótese H_1 , foi considerado o sinal “ \neq ”; portanto o valor de p deve corresponder as duas caudas da distribuição (BARBETTA, 2002).

Complementando, a regra $p (t \leq t_{\text{crítico}}) = 0,69$ e o nível de significância de $\alpha=0,05$ indicam que H_0 não é rejeitada. Em outras palavras, não há diferença significativa entre as médias. Logo, os dois Estudos de Caso foram realizados e o constructo andragógico não precisa ser alterado, pois foi bem desenvolvido em ambos os estudos.

Em suma, como não existe diferença significativa das médias entre as amostras populacionais, no passo seguinte os sujeitos foram reorganizados em dois grupos: brasileiros e uruguaios. Novamente, os dados foram analisados e a Tabela 2 mostra os dados descritivos.

Tabela 2 – Dados Descritivos de BR e UY

	Total	Média Final	Desvio Padrão (DP)	Coefficiente de Variação (CV)	Min	Max	Mediana	Moda
BR	12	58,08	17,65	30%	24,31	83,33	59,01	72,22
UY	15	65,02	16,63	26%	36,11	88,9	71,53	77,78

Fonte: Os autores (2020).

Esses dados (Tabela 2) mostram que a média da avaliação objetiva dos sujeitos uruguaios é maior que a dos brasileiros (em 11%), bem como o maior valor obtido (Max). A mediana e a moda desses sujeitos também são superiores. Esses valores estão refletidos no Gráfico 2, no qual o número de sujeitos considerados ACoD é maior para o grupo do UY. Pode-se observar, ainda, que os valores que mais diferem entre os grupos são o mínimo (Min) e a mediana.

Calculando o teste t de *Student* para as duas amostras independentes e assumindo que as hipóteses são

H0: $\mu_{BR} = \mu_{UY}$;

H1: $\mu_{BR} \neq \mu_{UY}$,

obteve-se que o valor de t é igual a -1.04 e o valor crítico de t (bicaudal) é de $2,05$. Então, o valor de t (estatístico) é menor que o t crítico e $P(T < =t)$ é igual a $0,30$. Logo, este valor é maior que o nível de significância de $\alpha=0,05$; portanto a H_0 não é rejeitada. Consequentemente, não existe uma diferença significativa das médias populacionais entre os sujeitos brasileiros e uruguaios.

Em síntese, a partir dos resultados pode-se concluir que o MRPC foi aplicado de forma igual em ambos os estudos de caso, pois as amostras não apresentaram diferença significativa entre as médias dos sujeitos envolvidos. Ainda, não há diferença significativa entre o desenvolvimento do PC da amostra de brasileiros e uruguaios.

CONSIDERAÇÕES FINAIS

Os resultados da pesquisa realizada evidenciam, portanto, que esta tese contribuiu para definir um modelo de referência de pensamento computacional e um constructo andragógico, tanto de ensino quanto de aprendizagem, que, aliados, classificam o sujeito ACod e avaliam o seu processo de construção do PC. A partir dos documentos desenvolvidos, o objetivo deste trabalho foi alcançado para cada aluno de forma individual. Desse modo, o objetivo específico – “analisar as habilidades e competências dos sujeitos com formação no Brasil/com formação no Uruguai, numa situação de ensino e aprendizagem de PC” – foi minuciosamente respondido.

É importante pontuar que, durante o desenvolvimento desta pesquisa, as ideias de David Ausubel e Malcolm Knowles estiveram sempre contempladas e foram as grandes influenciadoras deste processo. Já na escolha da ferramenta mostrou-se, na prática, o domínio dos conceitos em consonância com o pensamento de Knowles, Holton e Swanson (2011), como as escolhas dos temas de programação dos Apps, os quais devem estar sempre conectados com a realidade ou com o contexto dos aprendizes. A habilidade Compreensão foi influenciada por Ausubel (1968), ao afirmar que se deve sempre partir do que o aluno já sabe, complementado por testes, que devem avaliar diferentes aspectos do aprendiz, sendo um deles a compreensão, bem como aplicar os testes em formatos diferentes do que foram trabalhados em aula, a fim de verificar se a aprendizagem, de fato, ocorreu ou se o aluno apenas decorou as respostas.

A decisão de não definir previamente o projeto final, ou seja, negociar com os sujeitos qual o problema a ser resolvido, embasou-se em Knowles (1981), ao afirmar que o adulto sabe o que quer, tem prontidão e motivação para aprender, além de torná-los participantes do processo. A validação dos instrumentos propostos foi realizada nos dois Estudos de Caso apresentados. O MRPC pode ser replicável em outros estudos de caso ou como referência numa disciplina de PC na Graduação, considerando, porém, o tema do projeto final. Também pode ser ampliado para se tornar um metamodelo, no qual o professor poderá escolher a linguagem de programação e reprogramar os testes (avaliação objetiva) de acordo com a escolha.

A fim de ressaltar a importância do tema, é relevante citar as palavras de Rushkoff (2012, texto digital, tradução nossa): “Agora que temos computadores, estamos aprendendo a usá-los, mas não como programá-los. Quando não somos alfabetizados em código, devemos aceitar os dispositivos e *softwares* que usamos com quaisquer limitações que seus criadores tenham embutido neles”. Esse autor propõe uma reflexão sobre o

valor da alfabetização em código, independentemente da idade, para o exercício da plena cidadania, bem como da vida pessoal e social. É libertar o sujeito para definir as suas necessidades e executá-las, utilizando a tecnologia; em outras palavras, é passar da posição de consumidor para produtor de tecnologia.

Espera-se que esta pesquisa tenha contribuído para favorecer uma reflexão sobre a necessidade de os adultos serem alfabetizados em código para o exercício pleno da cidadania, bem como para uma definição e classificação da ACod e para uma nova habilidade para desenvolver o PC – Compreensão. A educação de adultos pode e talvez deva ser um *continuum* educativo, coextensivo à vida e ampliado às dimensões de uma sociedade que se modifica constantemente, o que ficou mais que evidente neste ano pandêmico.

AGRADECIMENTO

Gratidão por todo o apoio financeiro oferecido pelo Instituto Federal de Ciência e Tecnologia – Ifsul.

REFERÊNCIAS

- ARAÚJO, Ana Liz Souto Oliveira de. *Quantifying Computational Thinking Abilities*. 2019. Tese (Doutorado em Ciência da Computação) – Universidade Federal de Campina Grande, Campina Grande, 2019.
- AUSUBEL, David P. *Educational psychology: a cognitive view*. New York: Holt, Rinehart and Winston, 1968. *E-book*.
- AUSUBEL, David P. *Aquisição e retenção de conhecimentos: uma perspectiva cognitiva*. Tradução Lígia Teopisto. 1. ed. Rio de Janeiro: Plátano Edições Técnicas, 2003. *E-book*.
- AUSUBEL, David P.; NOVAK, Joseph Donald; HANESIAN, Helen. *Psicología Educativa: un punto de vista cognoscitivo*. 2. ed. México: Trillas, 1983. *E-book*.
- AVILA, Christiano et al. *Metodologias de avaliação do pensamento computacional: uma revisão sistemática*. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE), 28., 2017. Recife: SBC, 2017. p. 113. Disponível em: <https://doi.org/10.5753/cbie.sbie.2017.113>. Acesso em: 10 fev. 2018.
- BARBETTA, Pedro Alberto. *Estatística aplicada às Ciências Sociais*. 5. ed. Florianópolis, SC: UFSC, 2002. *E-book*.
- BRACKMANN, Christian Puhlmann. *Desenvolvimento do pensamento computacional através de atividades desplugadas na Educação Básica*. 2017. Tese (Doutorado em Informática na Educação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017. Disponível em: <https://doi.org/10.1111/j.1469-7610.2010.02280.x>
- BRENNAN, Karen; RESNICK, Mitchel. New frameworks for studying and assessing the development of computational thinking. *American Educational Research Association – Aera*, Washington, EUA, p. 1-25, 2012. Disponível em: https://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf. Acesso em: 10 maio. 2018.
- BROWN, Wayne. *Introduction to algorithmic thinking*. 2015. Disponível em: [https://raptor.martincarlisle.com/Introduction to Algorithmic Thinking.doc](https://raptor.martincarlisle.com/Introduction%20to%20Algorithmic%20Thinking.doc).
- CAMPOS, Geraldo Maia. *A escolha do teste mais adequado*. 2000. Disponível em: http://www.forp.usp.br/restauradora/gmc/gmc_livro/gmc_livro_cap14.html. Acesso em: 1º abr. 2020.
- CERF, Vinton G. The Science in Computer Science. Computer science revisited. *Communications of the ACM*, New York, p. 4, 2012. Disponível em: <https://doi.org/10.1145/2380656.2380658>. Acesso em: 12 ago. 2018.
- CORDENONZI, Walkiria Helena et al. Alfabetização – uma evolução do conceito: alfabetização e letramento em código. *Texto Livre: Linguagem e Tecnologia*, Belo Horizonte, MG, v. 13, n. 1, p. 137, 2020. Disponível em: <https://doi.org/10.17851/1983-3652.13.1.137-155>. Acesso em: 26 maio 2020.

- CORDENONZI, Walkiria Helena. *O desenvolvimento do pensamento computacional e as evidências da alfabetização em código em adultos*. 2020. Tese (Doutorado em Ensino) – Universidade do Vale do Taquari – Univates, Lajeado, 2020.
- CRTC. *Currículo de Referência em Tecnologia e Computação*. São Paulo, SP: [s.n.], 2018. Disponível em: <http://curriculo.cieb.net.br/>. Acesso em: 19 mar. 2019.
- FROTA, Vitor Bremgartner da. *Arcabouço conceitual de adaptação de recursos educacionais*. 2017. Tese (Doutorado em Informática) – Universidade Federal do Amazonas, Amazonas, 2017.
- GARCIA, Lenise Aparecida Martins. Competências e habilidades: você sabe lidar com isso. *Educação e Ciência on-line*, Brasília, 2005. Disponível em: http://www.educacao.es.gov.br/download/roteiro1_competenciasehabilidades.pdf. Acesso em: 15 maio 2020.
- GATTI, Bernardete A. Estudos quantitativos em educação. *Educação e Pesquisa*, São Paulo, v. 30, n. 1, p. 11-30, 2004. Disponível em: <https://doi.org/10.1590/s1517-97022004000100002>
- GIL, Antonio C. *Métodos e técnicas da pesquisa social*. 6. ed. São Paulo: Atlas, 2008. *E-book*.
- GROVER, Shuchi; PEA, Roy. Computational Thinking in K-12. *Educational Researcher*, Thousand Oaks, EUA, a. 42, n. 1, 1º jan. 2013, p. 38-43. Disponível em: <https://doi.org/10.3102/0013189X12463051>. Acesso em: 19 mar. 2019.
- KALELIOGLU, Filiz; GÜLBAHAR, Yasemin; KUKUL, Volkan. A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, Riga, Letônia, v. 4, n. 3, p. 583, 2016. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0360131519301605>. Acesso em: 4 mar. 2019.
- KATAI, Zoltan; TOTH, Laszlo. Technologically and artistically enhanced multi-sensory computer-programming education. *Teaching and Teacher Education*, v. 26, n. 2, p. 244-251, 2010. Disponível em: <https://doi.org/10.1016/j.tate.2009.04.012>. Acesso em: 13 set. 2018.
- KNOWLES, Malcolm S. *The modern practice of adult education: from pedagogy to andragogy*. 2. ed. Nova York: Cambridge Book, 1981. *E-book*.
- KNOWLES, Malcolm S.; HOLTON, Elwood F.; SWAMSON, Richard A. *Aprendizagem de resultados*. 2. ed. São Paulo: Elsevier, 2011. *E-book*.
- KORDAKI, Maria. A drawing and multi-representational computer environment for beginners' learning of programming using C: Design and pilot formative evaluation. *Computers and Education*, v. 54, n. 1, p. 69-87, 2010. Disponível em: <https://doi.org/10.1016/j.compedu.2009.07.012>
- KORKMAZ, Özgen; ÇAKIR, Recep; ÖZDEN, M. Yaşar. A validity and reliability study of the computational thinking scales (CTS). *Computers in human behavior*, v. 72, p. 558-569, 2017. Disponível em: <https://doi.org/10.1016/j.chb.2017.01.005>
- KOTSOPOULOS, Donna et al. A pedagogical framework for computational thinking. *Digital Experiences in Mathematics Education*, v. 3, n. 2, p. 154-171, 2017. Disponível em: <https://doi.org/10.1007/s40751-017-0031-2>. Acesso em: 22 maio 2020.
- MORAES, Marcelo Botelho da Costa. *Análise multivariada aplicada à contabilidade*. São Paulo, 2016. Disponível em: https://edisciplinas.usp.br/pluginfile.php/1582605/mod_resource/content/1/AnáliseMultivariada-Aula01.pdf. Acesso em: 1º abr. 2020.
- MOREIRA, Marco Antonio. *Mapas conceituais e aprendizagem significativa*. 1. ed. São Paulo: Centauro, 2010. *E-book*.
- MORENO, Julián. Digital competition game to improve programming skills. *Educational Technology and Society*, v. 15, n. 3, p. 288-297, 2012. Disponível em: <https://eric.ed.gov/?id=EJ992538>. Acesso em: 2 fev. 2019.
- MORENO LEÓN, Jesús; ROBLES, Gregório; ROMÁN GONZÁLEZ, Marcos. Dr. Scratch: análisis automático de proyectos scratch para evaluar y fomentar el pensamiento computacional. *Revista de Educación a Distancia*, Murcia, Espanha, v. 46, n. 10, 2015. Disponível em: <https://doi.org/10.6018/red/46/10>. Acesso em: 15 dez. 2018.
- MORENO LEÓN, Jesús; ROMÁN GONZÁLEZ, Marcos; ROBLES, Gregório. *On computational thinking as a universal skill: A review of the latest research on this ability*. In: IEEE GLOBAL ENGINEERING EDUCATION CONFERENCE (EDUCON), 2018. Tenerife, Espanha: IEEE, 2018. p. 1.684-1.689. Disponível em: <https://doi.org/10.1109/EDUCON.2018.8363437>. Acesso em: 22 jan. 2019.
- MORETTO, Vasco P. Reflexões construtivistas sobre habilidades e competências. *Dois Pontos: Teoria & Prática em Gestão*, Belo Horizonte, MG, n. 42, p. 54-54, 1999.

OPMANIS, Mārtiņš; DAGIENE, Valentina; TRUU, Ahto. Task types at “beaver” contests. *In: Informatics in Secondary Schools: Evolution and Perspectives*. Vilnius, Lithuania: Institute of Mathematics and Informatics, 2006. p. 509-519. Disponível em: <https://www.bebbras.org/sites/default/files/documents/publications/Opmanis-2006.pdf>. Acesso em: 2 abr. 2019.

PAPERT, Seymour. *Logo: computadores e educação*. 1. ed. São Paulo: Editora Brasiliense, 1980. *E-book*.

PERRENOUD, Philippe. *Construir as competências desde a escola*. 1. ed. Porto Alegre: Artmed, 1999. *E-book*.

PERRENOUD, Philippe *et al.* *As competências para ensinar no século XXI: a formação dos professores e o desafio da avaliação*. Porto Alegre: Artmed, 2002. *E-book*.

RAABE, André *et al.* Um instrumento para diagnóstico do pensamento computacional. *In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (CBIE)*, 6., 2017. Recife: SBC, 2017. p. 1.172-1.181. Disponível em: <https://doi.org/10.5753/cbie.wcbie.2017.1172>. Acesso em: 25 fev. 2019.

RAZALI, Nornadiah; WAH, Yap Bee. Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests. *Journal of Statistical Modeling and Analytics*, v. 2, n. 1, p. 13-14, 2011. Disponível em: https://www.researchgate.net/publication/267205556_Power_Comparisons_of_Shapiro-Wilk_Kolmogorov-Smirnov_Lilliefors_and_Anderson-Darling_Tests. Acesso em: 1º abr. 2020.

RESNICK, Mitchel *et al.* Scratch. *Communications of the ACM*, [S. l.], v. 52, n. 11, p. 60, 2009. Disponível em: <https://doi.org/10.1145/1592761.1592779>. Acesso em: 15 ago. 2018.

RIBEIRO, Leila; FOSS, Luciana; CAVALHEIRO, Simone André da Costa. *Entendendo o pensamento computacional*. 2017. Disponível em: <http://arxiv.org/abs/1707.00338>. Acesso em: 18 mar. 2019.

RODRIGUES, Rivanilson da Silva *et al.* Análise dos efeitos do pensamento computacional nas habilidades de estudantes no ensino básico: um estudo sob a perspectiva da programação de computadores. *In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE)*, 26., 2015. Maceió: SBC, 2015. p. 121. Disponível em: <https://doi.org/10.5753/cbie.sbie.2015.121>. Acesso em: 4 fev. 2018.

ROMÁN GONZÁLEZ, Marcos. *Códigoalfabetización y Pensamiento Computacional en Educación Primaria y Secundaria: Validación de un Instrumento y Evaluación de Programas*. 2016. Tese (Doctorado En Educación) – Universidad Nacional de Educación a Distancia, Madrid, 2016.

ROMÁN GONZÁLEZ, Marcos; PÉREZ GONZÁLEZ, Juan Carlos; JIMÉNEZ FERNÁNDEZ, Carmen. Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in human behavior*, v. 72, p. 678-691, 2017. Disponível em: <https://doi.org/10.1016/j.chb.2016.08.047>. Acesso em: 4 abr. 2019.

RUIPÉREZ, Beatriz Ortega. *Pensamiento Computacional y Resolución de Problemas*. 2017. Tese (Doctorado en Psicología) – Universidad Autónoma de Madrid, Madrid, 2017. Disponível em: <https://repositorio.uam.es/handle/10486/683810>. Acesso em: 15 abr. 2019.

RUSHKOFF, Douglas. *Code Literacy: A 21st-Century Requirement*. 2012. Disponível em: <https://www.edutopia.org/blog/code-literacy-21st-century-requirement-douglas-rushkoff>. Acesso em: 15 ago. 2018.

SEITER, Linda; FOREMAN, Brendan. Modeling the learning progressions of computational thinking of primary grade students. *In: ANNUAL INTERNATIONAL ACM CONFERENCE ON INTERNATIONAL COMPUTING EDUCATION RESEARCH (ICER)*, 9., New York, USA: ACM Press, 2013. p. 59. Disponível em: <https://doi.org/10.1145/2493394.2493403>. Acesso em: 3 abr. 2019.

SILVA, André Luís Silva da; DEL PINO, José Claudio. *Metodologias de Ensino – No contexto da formação continuada de professores*. Curitiba, Paraná: Appris, 2019. *E-book*.

TANG, Xiaodan *et al.* Assessing computational thinking: A systematic review of empirical studies. *Computers and Education*, v. 148, p. 103-798, 2020. Disponível em: <https://doi.org/10.1016/j.compedu.2019.103798>. Acesso em: 3 jun. 2020.

TRIVIÑOS, Augusto N. S. *Introdução à pesquisa em Ciências Sociais – a pesquisa qualitativa em Educação*. São Paulo: Atlas, 2015. *E-book*.

WING, Jeannette M. Computational thinking. *Communications of the ACM*, v. 49, n. 3, p. 33-35, 2006. Disponível em: <https://doi.org/10.1145/1118178.1118215>. Acesso em: 18 mar. 2019.

WING, Jeannette M. Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, v. 25, n. 2, p. 7-14, 2017. Disponível em: <https://www.learntechlib.org/p/183466/>. Acesso em: 18 mar. 2019.