

Um Constructo Andragógico para Avaliação do Pensamento Computacional no Adulto

Walkiria Helena Cordenonzi¹, José Claudio Del Pino²

walkiriacordenonzi@ifsul.edu.br, josepino@univates.br

¹IFSUL, Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense, Câmpus Santana do Livramento. Av. Paul Harris 410. Santana do Livramento/RS. Brasil

²Univates, Universidade do Vale do Taquari. Av. Avelino Talini, 171 - Bairro Universitário, Lajeado/RS - Brasil

Resumo

Atualmente, muitos cidadãos têm acesso à internet e aos dispositivos móveis, como celulares, por exemplo, e os utilizam para executarem suas tarefas tanto particulares, quanto laborais. Estas mudanças compeliram os sujeitos a serem alfabetizados digitalmente. Porém, estes conhecimentos já não estão sendo suficientes, uma vez que os mesmos são encorajados a produzir tecnologia, ao invés de somente consumir da mesma, a fim de satisfazer o mercado laboral enquanto profissional, independente da sua área de atuação. Para um sujeito se tornar um agente produtor de tecnologia, preconiza-se que este desenvolva habilidades, pelo menos, de resolução de problemas, mais precisamente o Pensamento Computacional (PC) - habilidade desejável para cidadãos do século XXI. Este trabalho teve como objetivo compreender e analisar o desenvolvimento do PC nos sujeitos adultos e para isso propõe um método de avaliação do processo de ensino e aprendizagem de Pensamento Computacional. A partir do constructo andragógico inovador, embasado na Teoria da Aprendizagem Significativa e na Andragogia, no qual foram definidos um Modelo de Referência de Pensamento Computacional, o planejamento de um curso e um modelo de avaliação, obtém-se a classificação do sujeito como Alfabetizado em Código, Pensador Computacional Desplugado ou nenhum deles.

Palavras Chave: Avaliação do Pensamento Computacional. Andragogia. Aprendizagem Significativa. Pensamento Computacional.

Un constructo andragógico para evaluar el pensamiento computacional en adultos

Resumen

Hoy en día, muchos ciudadanos tienen acceso a Internet y a dispositivos móviles, como teléfonos celulares, por ejemplo, y los utilizan para realizar sus tareas privadas y laborales. Estos cambios obligaron a los sujetos a ser alfabetizados digitalmente. Sin embargo, este conocimiento ya no es suficiente, ya que se les anima a producir tecnología, en lugar de solo consumirla, para satisfacer al mercado laboral como profesional, independientemente de su área de especialización. Para que un sujeto se convierta en un agente productor de tecnología, se recomienda que desarrolle al menos habilidades para la resolución de problemas, más precisamente el Pensamiento Computacional (PC), una habilidad deseable para los ciudadanos del siglo XXI. Este trabajo tuvo como objetivo comprender y analizar el desarrollo de la PC en sujetos adultos y para ello propone un método de evaluación del proceso de enseñanza y aprendizaje del Pensamiento Computacional. Partiendo de el innovador constructo andragógico, basado en la Teoría del Aprendizaje Significativo y la Andragogía, en el que se definieron un Modelo de Referencia del Pensamiento Computacional, la planificación de un curso y un modelo de evaluación, se obtiene la clasificación del sujeto como Alfabetizado en Código, Pensador Computacional Desconectado o ninguno de los anteriores.

Palabras clave: Evaluación del pensamiento computacional. Andragogía. Aprendizaje significativo. Pensamiento computacional.

An Andragogical Construct for Assessing Computational Thought in Adults

Abstract

Today, many citizens have access to the internet and mobile devices, such as cell phones, for example, and use them to perform their private and work tasks. These changes compelled the subjects to be digitally literate. However, this knowledge is no longer sufficient, since they are encouraged to produce technology, instead of just consuming it, in order to satisfy the labor market as a professional, regardless of their area of expertise. For a subject to become a technology-producing agent, it is recommended that he develop at least problem-solving skills, more precisely Computational Thinking (PC) - a desirable skill for 21st century citizens. This work aimed to understand and analyze the development of CP in adult subjects and for that purpose proposes a method of evaluating the teaching and learning process of Computational Thinking. Based on the innovative andragogical construct, based on the Theory of Meaningful Learning and Andragogy, in which a Reference Model of Computational Thinking was defined, the planning of a course and an evaluation model, the classification of the subject as Literate in Code, Unplugged Computational Thinker or none of them.

Keywords: Computational Thinking Assessment. Andragogy. Meaningful Learning. Computational Thinking.

Un andragogique construit pour évaluer la pensée informatique chez les adultes

Résumé

Aujourd'hui, de nombreux citoyens ont accès à Internet et aux appareils mobiles, tels que les téléphones portables, par exemple, et les utilisent pour effectuer leurs tâches privées et professionnelles. Ces changements ont obligé les sujets à maîtriser le numérique. Cependant, ces connaissances ne sont plus suffisantes, car elles sont encouragées à produire de la technologie, au lieu de simplement la consommer, afin de satisfaire le marché du travail en tant que professionnel, quel que soit leur domaine d'expertise. Pour qu'un sujet devienne un agent de production de technologie, il est recommandé qu'il développe au moins des compétences en résolution de problèmes, plus précisément la Pensée Computationnelle (PC) - une compétence souhaitable pour les citoyens du 21^e siècle. Ce travail visait à comprendre et à analyser le développement de la PC chez l'adulte et à cet effet propose une méthode d'évaluation du processus d'enseignement et d'apprentissage de la pensée computationnelle. La proposition de la construit andragogique innovante, basée sur la théorie de l'apprentissage significatif et de l'andragogie, dans laquelle un modèle de référence de la pensée computationnelle, la planification d'un cours et un modèle d'évaluation ont été définis, obtient la classification du sujet comme alphabétisé en code, Penseur informatique débranché ou aucun d'entre eux.

Mots clés: Évaluation de la pensée informatique. Andragogie. Apprentissage significatif. Pensée informatique.

1. INTRODUÇÃO

A sociedade, ao longo do tempo de sua história, vai evoluindo e se transformando. Sua dinâmica foi remodelada devido às Tecnologias Digitais de Informação e Comunicação (TDIC). Estas tecnologias exercem constantes e permanentes adaptações na vida cotidiana e, com respeito às mudanças, não há mais retorno. Sendo assim, tem-se uma sociedade permeada pelo mundo digital. A partir dessas transformações, gerou-se uma onda de ajustes e acomodações nesta comunidade. Isso significa que já se está percebendo os reflexos nas tarefas diárias, de tal sorte que uma atividade simples já não se faz da mesma maneira como se fazia há um tempo.

Desse modo, aos cidadãos, cabe a busca do conhecimento e, juntamente, a alfabetização digital (AD), pois a exploração e a descoberta destas informações são atividades chave do dia a dia. Segundo a definição que consta no livro Sociedade da Informação no Brasil - Livro

Verde¹ a AD pode ser definida como “a aquisição de habilidades básicas para o uso de computadores e da Internet”. Ademais, é impulsionar os sujeitos para que se tornem agentes capazes de usar as tecnologias e aprimorando sua capacidade laboral para aumentar sua eficiência e eficácia, além de desenvolver sua capacidade de aprender a aprender.

Wing, em 2006, apresenta um novo conceito: pensamento computacional (PC). Este pode ser entendido como o processo de resolver e identificar problemas, utilizando métodos e técnicas provenientes da ciência da computação. Ainda, afirma esta autora, que ao utilizar-se destas técnicas facilitará a obtenção dos resultados, principalmente para encontrar soluções de problemas complexos. Não somente esta interpretação sobre PC, como também Wing sugere ser compreendido como um

¹ Disponível em: <http://www.socinfo.org.br>. Acesso em 05 mar 2018.

conjunto de habilidades e apropriação de determinados processos cognitivos (Wing, 2017). Na literatura, encontram-se diferentes definições sobre este termo (Brackmann, 2017; ISTE & CSTA, 2011; Román González, 2016), e em virtude disso, ainda hoje não há um consenso sobre o mesmo. De tal forma, o PC e suas habilidades correspondentes, são indispensáveis para os profissionais da área de Ciência da Computação. No entanto, Wing afirma que estas habilidades são proveitosas para todas as pessoas, independente da profissão ou domínio de conhecimento. Sobretudo, para facilitar o empoderamento dos profissionais quando submetidos a resolver problemas em qualquer área, mesmo que aparentemente estas não estejam relacionadas.

Com relação a computação, a partir de 2012, já está sendo considerada como ciência pela *Association for Computing Machinery* (ACM) (Cordenonzi, 2020).

Enquanto não se adequam os currículos para auxiliar o desenvolvimento do PC, muitos investigadores propuseram cursos, nos quais desenvolvem atividades neste sentido, alguns destes focando na programação, outros em atividades desplugadas (Brackmann, 2017). É certo que para programar é necessário o desenvolvimento de certas habilidades, as quais agregam na aprendizagem de algoritmos e, ainda, uma linguagem de programação. Em outras palavras, desenvolver a PC na programação não significa apenas programar, mas sim desenvolver todo o processo para encontrar a solução de um problema. Desse modo, une-se a preocupação do processo de ensino e aprendizagem, para analisar como os sujeitos adultos desenvolvem seu pensamento computacional, tanto para atender as demandas da sociedade, quanto para lograr resolver problemas do cotidiano. Assim sendo, associa-se a preocupação das mudanças causadas pelo advento das tecnologias, no qual o sujeito adulto irá ou já está no mercado de trabalho e concomitantemente buscando sua qualificação.

Esta pesquisa tem por objetivo compreender e analisar o desenvolvimento do PC nos sujeitos adultos (entendidos como aqueles que já completaram a educação básica). Para tanto, apresenta um constructo andragógico, composto por três partes: um modelo de referência para pensamento computacional (MRPC), um método de avaliação e um protocolo de curso para a execução e validação da proposta. Como resultado obtém-se a classificação dos sujeitos em uma das seguintes categorias (Cordenonzi, 2020): Alfabetizado em Código (ACod), pensador computacional desplugado (PCD) ou nenhum destes. Este constructo foi aplicado em dois estudos de casos, detalhados em Cordenonzi (2020) e foi possível validar este modelo de forma prática. Este artigo, primeiramente, apresenta o entendimento sobre PC, seguindo sobre a descrição de trabalhos correlatos. Na seção 4, as teorias de aprendizagem norteadoras desta pesquisa estão explanadas e, na sequência, o constructo andragógico. Posteriormente, na seção 6, as considerações finais são discorridas. Já na seção 7 as referências consultadas estão listadas.

2. PENSAMENTO COMPUTACIONAL

Há diversos entendimentos e definições acerca do conceito de PC (do inglês, *Computational Thinking*), porém ainda não há um consenso sobre o que é PC, ou o que ele implica. Neste sentido, Wing, em 2017, redefiniu o conceito como: “O pensamento computacional é 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” (Wing, 2017, p. 8). Além disso, a autora acrescenta que é uma forma de pensamento recursivo, usando a abstração e a decomposição, ou seja, é fazer uso do raciocínio heurístico para encontrar a solução. Ainda, defende que é uma habilidade fundamental, a qual poderá ser utilizada por todas as pessoas do século XXI, mostrando-se semelhante à de ler, escrever e calcular.

Partindo do exposto por Wing, o PC refere-se às pessoas e como estas desenvolvem a capacidade de resolver e propor problemas, independentemente dos recursos computacionais. Esta autora sugere que todas as universidades deveriam ter no currículo uma disciplina como “Formas de pensar como um cientista da computação”, não somente iniciando com atividades e tarefas de programação, mas para todos, independentemente do curso ou idade do sujeito.

A abstração pode ser considerada como uma das mais importantes habilidades, sendo indiscutível sua importância para o desenvolvimento do PC (Grover & Pea, 2013). Não se pode confundir PC com programação, pois esta é a automação das abstrações do indivíduo (Wing, 2017). Reforçando sobre as habilidades mais importantes, Avila *et al.* (2017) analisaram 58 artigos e chegaram aos mesmos resultados: as habilidades mais trabalhadas foram pensamento algorítmico, resolução de problemas e abstração. Kalelioglu *et al.* (2016) analisaram 125 artigos e afirmam que sobre o levantamento dos conceitos mais utilizados para descrever o significado de PC, os mais citados são resolução de problemas e abstração, nesta ordem. Portanto, não há consenso sobre a definição e a forma de avaliação de PC. Torna-se emergente que métodos avaliativos sejam aplicados, a fim de que se possa conhecer os resultados do desenvolvimento do PC, mediante as mudanças curriculares que estão sendo propostas e já implementadas por vários países e instituições de ensino (ISTE & CSTA, 2011).

No Brasil, após realizada uma pesquisa no Catálogo de Teses e Dissertações da Capes, em abril/2020, figuraram 72 dissertações de mestrado que continham as palavras “pensamento computacional”. Porém, somente em 18% dos estudos foram realizados em cursos superiores (adultos). Pode-se perceber que o interesse da maioria dos pesquisadores está na educação básica. Essa tendência, com relação à faixa etária, no intervalo entre 6 a 18 anos, acompanha as publicações internacionais. Das 15 teses de doutorado analisadas, no mesmo Portal, apenas 5 teses tiveram como sujeitos cursantes no Ensino Superior, que

corresponde a 26,66%. Quanto as teorias de aprendizagem citadas, tem-se que 57.14% utilizaram o construcionismo e somente uma citou o construtivismo de Piaget. Já a aprendizagem significativa aparece apenas em dois trabalhos, um no nível fundamental e outro no superior e em nenhuma dissertação. Sobre a teoria de aprendizagem de adultos, proposta por Knowles, Holton e Swanson (2011), utilizada neste trabalho não foi encontrada em nenhuma referência.

3. TRABALHOS CORRELATOS

Este tópico tem por objetivo discorrer sobre a preocupação da comunidade científica em avaliar o PC. Após a primeira publicação de Wing, ocorrida em 2006, muitos trabalhos têm sido desenvolvidos na área de pensamento computacional (PC), com diferentes abordagens. Desde 2011 (Brackmann, 2017), os currículos de alguns países têm sido modificados, a fim de desenvolver nos estudantes, com idades que variam entre 4 e 14 anos, habilidades para que adquiram o *status* de “pensador computacional” (“*computational thinker*”) (Brennan & Resnick, 2012; ISTE & CSTA, 2011).

Não há um padrão ou modelo genérico para a mensuração do PC. Por isso, torna-se emergente que métodos avaliativos sejam aplicados, a fim de que se possa conhecer os resultados do desenvolvimento do PC, mediante as mudanças curriculares que estão sendo propostas e já implementadas, ou mesmo em intervenções isoladas.

A seguir, estão descritos alguns trabalhos encontrados na literatura sobre avaliação de PC. Korkmaz, Çakir e Özden (2017) definiram uma escala que determina os níveis de Habilidade de Pensamento Computacional (*computational thinking skills*) e aplicaram em alunos da Amasya University, na Turquia. Esta escala está dividida em 29 itens, distribuídos em fatores (criatividade, pensamento algorítmico, cooperação, pensamento crítico e resolução de problemas). Aplicaram para 1306 estudantes e concluíram que, estatisticamente, este método é válido e confiável. Este trabalho é uma das poucas referências sobre avaliação do 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.

O grupo de Koh *et al.* (2010) desenvolveu uma ferramenta denominada de CTPG (*Computational Thinking Pattern Graph*) para a avaliação semântica de jogos e simulações, desenvolvidos por estudantes, que analisa os conceitos de pensamento computacional implementado pelos alunos. Aplicaram em vários grupos de estudantes universitários, justificando que as linguagens visuais não são adequadas para se obter de forma clara o tipo de conhecimento do qual o aluno se apropriou.

Uma experiência desenvolvida na Universidade de Montana (Van Dyne & Braun, 2014) descreve um curso

sobre PC (que possui o mesmo nome), para alunos matriculados no Curso de Ciência da Computação. A finalidade foi de melhorar a resolução de problemas de forma analítica e obtiveram sucesso nos resultados. Concluíram que o curso teve um impacto significativo na melhoria das habilidades de resolução de problemas. Para avaliação, utilizaram o *Whimbey Analytical Skills Inventory* (WASI)² em dois momentos, como pré e pós testes.

Um teste realizado para avaliar o desenvolvimento das habilidades do PC foi desenvolvido e aplicado em estudantes do primeiro semestre de um Curso de Computação na Universidade de Rhodes (África do Sul). O planejamento seguiu a Taxonomia de Bloom e o teste foi aplicado para avaliar os saberes dos estudantes antes de os alunos iniciarem um conhecimento formal dos conteúdos. As questões foram retiradas de *Computer Olympiad Talent Search*³. Concluíram que o PC não é uma habilidade que possa ser desenvolvida nos alunos, antes deve ser melhor abordada. Sendo assim, sugerem que mais estudos deveriam ser realizados (Gouws *et al.*, 2013).

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. Porém, não consideram em sua pesquisa o nível de ensino nos quais os testes foram aplicados, tampouco diferenciaram a avaliação realizada com o uso de ferramentas. Cabe ressaltar que quando os testes são realizados por automação, eles ficam limitados ao *software* utilizado para o desenvolvimento da solução do problema, como, por exemplo, o teste *Dr. Scratch* (descrito a seguir).

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

O primeiro foi proposto na tese de doutorado de Román González (2016), nomeado de *Test de Pensamiento Computacional (CT-Test)*, foi projetado, especificamente, para a educação espanhola que cursa o final do ciclo básico e secundário (idade do aluno entre 10 e 16 anos ou seu correspondente K-12)⁴. Este mede o nível de desenvolvimento do PC por meio de um questionário

² Desenvolvido por Arthur Whimbey e Jack Lochhead, para avaliar e desenvolver capacidades cognitivas para a resolução de problemas.

³ Disponível em: <http://www.olympiad.org.za/talent-search>. Acesso em: 3 ago. 2018.

⁴ No Brasil corresponde ao ensino básico, nos EUA a “middle school”.

composto por 28 questões objetivas, com opções de 4 respostas (somente uma correta). Concluiu que o PC não só é uma habilidade cognitiva de solução de problemas, como é independente de sexo. Assinalaram, ainda, que o teste é limitado porque não é possível avaliar e medir processos cognitivos como “aplicar” e “assimilar” o PC.

*Dr. Scratch*⁵ é uma aplicação *Web* que analisa automaticamente os projetos desenvolvidos no *Scratch*. Fornece *feedback* que permite ao projetista melhorar seu código, aprender com seus erros e desenvolver suas habilidades de PC (Moreno León *et al.*, 2018). No entanto, esta ferramenta limita-se às atividades desenvolvidas exclusivamente no ambiente *Scratch* e não oferece suporte para abstração de funções e procedimentos no código (Grover & Pea, 2013). O *Scratch* foi desenvolvido para ser utilizado entre as idades de 8 e 16 anos, embora seus criadores afirmam que pode ser empregado em qualquer idade (Brennan & Resnick, 2012). Como resultado, esta ferramenta apresenta um retorno que indica a qualidade do programa desenvolvido no *Scratch*, sendo considerada uma ferramenta de avaliação formativa (Román González *et al.*, 2017).

O *International Challenge on Informatics and Computational Thinking*⁶, denominado de *Bebras* (em lituano significa Castor), é uma iniciativa internacional para promover a computação e pensamento computacional entre os alunos em todas as idades. É um desafio que consiste em um conjunto de questões (chamadas de tarefas) que tem por finalidade testar o nível de desenvolvimento das habilidades do PC (Dagiené *et al.*, 2017), entre elas: a resolução de problemas, decomposição, projeto de algoritmos, reconhecimento de padrões, generalização e abstração. As tarefas são a parte essencial do desafio e cada uma delas pode abordar conceitos de informática ou de PC (Dagiené *et al.*, 2017). Os desafios são separados por idade: *Pre-Primary* (entre 5 e 8 anos), *Primary* (8 e 10), *Benjamin* (10 e 12), *Cadets* (12 e 14), *Junior* (entre 14 e 16) e *Senior* (idade acima de 16 anos). Os participantes são geralmente supervisionados por professores que podem integrar o desafio *Bebras* em suas atividades de ensino. Em 2018, o *Bebras* foi aplicado com 2.614.000 participantes de 45 países. No entendimento de Moreno León, Román González e Robles (2018), este teste pode ser aplicado para qualquer faixa etária.

O *Bebras* está sendo citado neste contexto, pois muitas questões que o compõem foram utilizadas nos testes avaliativos que constituem o constructo andragógico desta tese. Por outro lado, é importante ressaltar que esta prova surgiu para ser utilizada em competições, e não como um método de avaliação de PC. Portanto, ao ser utilizado, está medindo habilidades diferentes do *CT-Test* (Wiebe *et al.*, 2019), visto que, neste último, suas questões são

baseadas, mais especificamente, em programação de blocos.

Muitas outras pesquisas sobre APC foram encontradas (Kordaki, 2010; Seiter & Foreman, 2013), mas não estão descritas, pois os sujeitos das pesquisas não são considerados adultos.

Já as pesquisas de Lye e Koh (2014) e de Cavalcante, Costa e Araújo (2016) e Araújo (2019) são mencionadas, visto que ambas utilizaram no seu processo de avaliação o *framework* de Brennan e Resnick (2012) (que também foi fonte para a proposta de avaliação desta pesquisa). Ambos os trabalhos abrangeram o *framework* nas 3 dimensões – conceito, prática e perspectiva. O primeiro, de Lye e Koh (2014), realizaram seus olhares no nível K-12. Concluíram que, a partir da resolução de problemas e utilizando-se do construcionismo para desenvolver um artefato concreto, auxiliou no desenvolvimento do PC. Já Cavalcante, Costa e Araújo (2016), utilizaram o *framework* em crianças de EF (Ensino Fundamental) e EM (Ensino Médio), utilizando a plataforma Code.org e programação em blocos.

Em síntese, o campo da avaliação do pensamento computacional (APC) ainda está carente de pesquisas. Foi possível analisar que boa parte das metodologias propostas são do tipo qualitativas, que há poucas ferramentas para avaliação que produzem *feedback* imediato e são restritas ao ambiente de desenvolvimento utilizado. Ainda mais que o desenvolvimento de metodologias de aplicação e avaliação do PC para crianças e jovens (que cursam o ensino básico) contam com um maior referencial teórico e prático se comparado ao nível universitário ou adultos, afirmam Tang *et al.* (2020). Na pesquisa realizada por esses autores, a partir de 77 artigos anteriores a agosto de 2019, apenas 15% estão focadas no Ensino Superior e concluem que ainda há poucas publicações na área de PC, sugerindo a integração de várias ferramentas para melhorar a avaliação da aprendizagem sobre PC.

Cabe esclarecer que o método proposto por Brennan e Resnick (2012), de avaliação do desempenho do PC, não foi utilizado neste trabalho, porque é dependente da ferramenta *Scratch*.

Outro ponto a ser considerado, com relação às teorias de aprendizagem relacionadas na literatura já citada, é que somente Lye e Koh (2014) basearam seu trabalho no construcionismo de Papert, os demais autores não fizeram referências.

A seguir, as teorias de aprendizagem que fundamentaram este trabalho são discorridas.

4. TEORIAS DE APRENDIZAGEM

Conforme já se vem observando, o desenvolvimento do PC, embora remeta a preocupação desde o século anterior, nos últimos 20 anos está tomando forma, exigindo mudanças de pensamento e atitudes do

⁵ Disponível em: <http://www.drscratch.org/>. Acesso em: 16 maio 2018.

⁶ Disponível em: <https://www.bebas.org/>. Acesso em: 10 jul. 2018.

professor, do aluno, do currículo e dos materiais instrucionais. Para subsidiar o constructo andragógico desta pesquisa, foram consideradas a Teoria de Aprendizagem Significativa (Ausubel, 2003) e a Andragogia (Knowles *et al.*, 2011), ambas descritas resumidamente a seguir:

Ausubel (2003) propõe uma teoria cognitivista que se preocupa com os mecanismos internos da mente, cujo principal conceito desta teoria é a aprendizagem significativa. O sujeito 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, 2003). Esta ancoragem é resultante de uma situação consciente, diferente e idiossincrática de cada aprendiz e é dependente dos seus subsunçores, do nível de desenvolvimento do aprendiz, da sua habilidade intelectual e do modo como é exibido o material instrucional. Importante destacar que a assimilação ausubeliana é um processo ativo do ponto de vista cognitivo que acontece no aprendiz.

A palavra Andragogia é formada a partir dos vocábulos gregos *andros* (homem) + *agein* (conduzir) + *logos* (tratado, ciência) (Knowles *et al.*, 2011). Fazendo referência à ciência da educação de adultos, a Andragogia faz oposição à Pedagogia, proveniente igualmente dos termos gregos *paidós* (criança) + *agein* (conduzir) + *logos* (tratado ou ciência), referindo-se à ciência da educação de crianças. Em 1981, Knowles, Holton e Swanson (2011) 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”. A Andragogia deve ser entendida como a filosofia, a ciência e a técnica da educação de adultos. Estes pesquisadores ainda complementam que os adultos e as crianças aprendem de formas diferentes. A fase adulta traz a maturidade e a independência, ela torna os indivíduos capazes de analisar, criticar, avaliar situações, tomando em conta suas experiências. Nesta fase, o importante é experienciar e vivenciar os momentos para que se tornem significativos, ou seja, o aprender fazendo.

A seguir o constructo andragógico desenvolvido é detalhado.

5. CONSTRUCTO ANDRAGÓGICO

Quanto às investigações na área de PC, seu entendimento e formas de avaliação ainda são muito recentes. Por isso, a comunidade acadêmica não formou um conceito ou entendimento comum sobre este tema. As pesquisas estão focadas no Ensino Básico, em sua grande maioria (Koh *et al.*, 2010). Por isso, esta proposta se torna importante e inovadora, sob vários aspectos, entre eles o uso da andragogia e o método avaliativo.

5.1. Proposição das Habilidades

Na literatura há muitas divergências sobre a definição de habilidades e competências. Em geral, as habilidades são consideradas como algo menos amplo do que as competências. Assim, a competência estaria constituída por várias habilidades. Entretanto, uma habilidade não “pertence” a determinada competência, uma vez que uma mesma habilidade pode contribuir para competências diferentes.

Habilidade é o saber-fazer, ou o conjunto de saberes, as capacidades, as informações, os recursos cognitivos, o conhecimento. Seguindo, então, a competência é a faculdade de mobilizar o saber-fazer a fim de “solucionar com pertinência e eficácia uma série de situações” (Perrenoud, 1999, p. 30). Em outras palavras, quando as habilidades são ativadas podem se tornarem competências, que por sua vez, teriam potencial para serem ancoradas na estrutura cognitiva. Portanto, no âmbito deste trabalho, o entendimento é de que as habilidades estão associadas ao saber fazer, ou seja, pode ser uma ação física ou mental que aponta para uma capacidade adquirida.

No contexto deste trabalho, sugere-se as seguintes habilidades (mínimas) que o sujeito deverá apresentar para desenvolver o PC.

A primeira delas é a ‘compreensão’. Acredita-se que este é o primeiro passo para a resolução de um problema, ou seja, a capacidade de entender o significado da informação, em oposição ao processamento meramente mecânico. Inicialmente, o sujeito precisa entender (compreender) o problema, quer dizer, o que está sendo solicitado e qual o contexto. A partir daí, ser capaz de significar e aplicar, estabelecendo uma organização hierárquica entre compreensão e a habilidade de abstração. Como resultado da falta de compreensão, o indivíduo não consegue resolver o problema, pois a partir da leitura não é capaz de identificar quais são as partes importantes (abstração).

Na taxonomia de Bloom, em sua edição revisada, a palavra ‘compreensão’ foi substituída por ‘entender’, mantendo o mesmo significado que é de relacionar uma conexão entre o novo e o conhecimento previamente adquirido, em que a informação consegue ser reproduzida pelo aprendiz, pois foi entendida. Portanto, a 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, a partir da leitura, o sujeito precisa criar uma representação mental do problema, sendo o mais objetivo possível. Isso quer dizer: ler e compreender o que foi lido, podendo utilizar-se de 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 para o passo seguinte, no qual deverá obter a solução do problema, e para isso necessitará de outras

habilidades (apresentadas a seguir, como, por exemplo, abstração e resolução de problemas).

Corroborando com o entendimento sobre a habilidade de compreensão, Roazzi e Hodges (2013, p. 4) discorrem que “compreender torna-se, portanto, uma atividade de construção, interativa e ativa, que requer a integração de informações novas, contidas não apenas no texto, mas também no interior das estruturas de conhecimento do sujeito que está lendo”. Pode-se perceber nesta definição a presença dos elementos da teoria da aprendizagem significativa.

Em suma, a ‘compreensão’ é uma habilidade proposta nesta tese, a fim de apoiar os constructos definidos, considerada de grande importância andragógica, através da qual se pode perceber o entendimento do sujeito frente à resolução de um problema, que é um dos fatores críticos para o desenvolvimento do PC.

A segunda habilidade é a ‘abstração’, que no Currículo de Referência em Tecnologia e Computação (CRTC) foi definida como envolvendo a “[...] filtragem dos dados e sua classificação, ignorando elementos que não são necessários, visando aos que são relevantes. Envolve também formas de organizar informações em estruturas que possam auxiliar na resolução de problemas” (CRTC, 2018, p. 19). Em 2006, Wing definiu ‘abstração’ como sendo o processo para decidir quais os detalhes que deverão ser destacados e quais descartados. Onze anos depois, a mesma autora complementa que a abstração 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, ficando, desta forma, a abstração como o próximo passo. A partir da abstração, pode-se determinar os pontos importantes e descartar os dados que não são relevantes, assim avançando para o próximo passo que é a resolução de problemas. A abstração é considerada importante e também difícil de ser desenvolvida, principalmente no tema que trata sobre resolução de problemas.

A habilidade ‘resolução de problemas’ centra-se na capacidade do sujeito de encontrar uma ou várias

soluções, independente do formato da sua resposta. Assim sendo, pode abarcar a decomposição de problema entendida, no CRTC como “o processo pelo qual os problemas são divididos em partes menores e mais fáceis de resolver” (CRTC, 2018, p. 19), quebrando em partes menores e gerenciáveis (ISTE & CSTA, 2011). Da mesma forma, e justificando sua importância, Ausubel descreveu que a capacidade de resolver problemas constitui a meta primária da educação.

Para entender a habilidade de ‘resolução algorítmica’, consideremos que 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. Portanto, essa habilidade está na prática do sujeito, a partir da habilidade anterior, na qual encontrou a solução do problema, traduzindo para um algoritmo ou *software*.

A ‘avaliação’ é a última habilidade proposta, a qual é entendida como a capacidade do sujeito após apresentar a resolução do problema através de algoritmo e/ou programa, de testar e avaliar a sua correção. Em outras palavras, é entender o que foi implementado, ou seja, é a aptidão de analisar um código-fonte ou executável quanto às suas saídas. Também, entende-se como a avaliação ou teste de uma proposta de solução de um problema apresentado, independente do seu formato.

A partir de uma organização lógica das habilidades, propõe-se um modelo de referência, descrito detalhadamente na seção seguinte.

5.2 Modelo de Referência para o desenvolvimento do PC – MRPC

Considera-se que as habilidades já citadas sejam as mínimas e necessárias a serem utilizadas para que o indivíduo seja capaz de resolver um problema e testá-lo. Na Figura 1, o modelo de referência para o desenvolvimento do pensamento computacional (MRPC) está representado de forma gráfica.

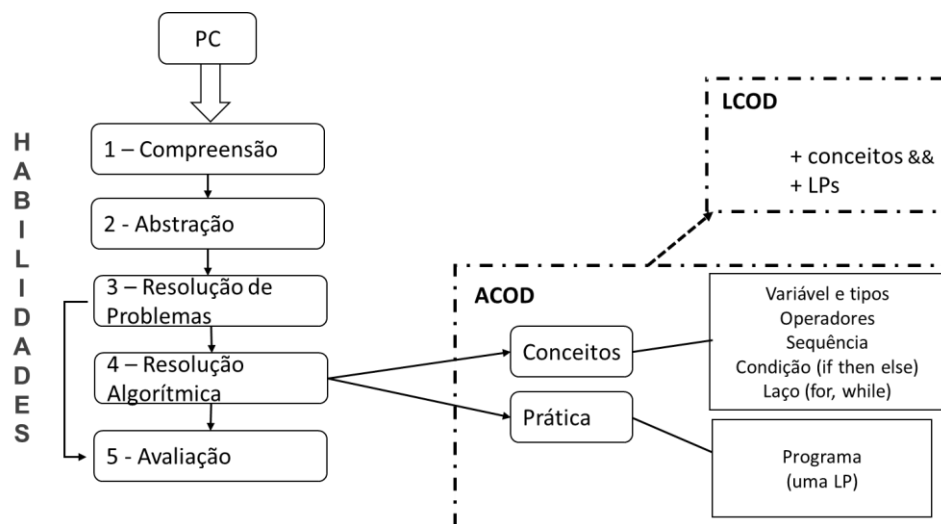


Figura 1 - Modelo MRPC
Fuente: autores (2019)

Acompanhando a Figura 1, pode-se perceber a existência de uma dependência lógica, sugerida e não obrigatória, para desenvolver o PC. Em outras palavras, a partir de um problema, a primeira habilidade que deve ser mobilizada é a compreensão, pois sem que o sujeito entenda o problema, não será capaz de resolvê-lo. Quanto ao que se entende como problema, este pode ser proposto por alguém ou percebido pelo próprio sujeito. Pois este tipo de pensamento (computacional) não está somente restrito a resolver um problema e também de definir uma situação que exige uma solução.

A partir da compreensão apropriada, a abstração passa a ser uma habilidade que naturalmente será mobilizada. Ou seja, o sujeito, compreendendo sua situação, espera-se que identifique os pontos importantes e descarte os quais não têm utilidade. Após esta organização, o sujeito passa a focar seus esforços em uma solução adequada. Neste ponto, não é importante quais métodos ou técnicas são utilizadas 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. Caso o caminho escolhido pelo sujeito seja o de transformar a sua solução em um algoritmo – Habilidade 4, necessita ancorar os conceitos mínimos definidos e descritos, a seguir:

- variável: posição de memória que pode ser identificada através de um nome e armazena um dado, em um determinado instante. Pode assumir vários tipos, dependendo da linguagem de programação (LP) utilizada, e seu valor pode ser alterado durante a programação.

- constante: definida igualmente como uma variável, com a diferença que seu valor não pode ser alterado durante a execução do programa.

- operadores:

- 1) matemáticos: soma, diferença, divisão, multiplicação, potência, entre outros;
- 2) lógicos: “e” (*and*) e “ou” (*or*);
- 3) relacionais: maior, menor, maior igual, menor igual, igual, entre outros.

- estrutura condicional: representado pelos comandos IF/THEN/ELSE.
- estruturas de repetição: representado pelos comandos FOR e WHILE.
- passos com sequência lógica baseados em eventos.

Porém, não basta apenas conhecer estes conceitos de programação e sim, também, 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 estes conceitos, é preciso que este algoritmo seja traduzido para uma linguagem de programação (LP), tornando-se então um software.

A partir do programa, ou do algoritmo, deve-se fazer os testes e 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, a partir de instrumentos avaliativos, infere-se para o sujeito o status de ACod, ou seja, está alfabetizado em código (*code literacy*). Segundo Cordenonzi (2020, p. 94):

“- Um indivíduo **Alfabetizado em Código** é aquele capaz de ler, interpretar e escrever um código-fonte e gerar um código executável (programa).

- Um indivíduo **Letrado em Código** é um ACod extrapolando suas habilidades e competências para codificar em diferentes linguagens (de programação) e novos conceitos.”

Cabe salientar que o reconhecimento do sujeito, neste estado, exige como pré-requisito a alfabetização digital. Porém, há de se considerar que o mesmo não pode ser atribuído com apenas poucas situações.

Em uma primeira situação, considerando que o sujeito/aprendiz nunca teve contato, ou seja, não apresenta nenhum subsunçor para a habilidade 4, e com programação, o processo de ensino e aprendizagem deve ser feito de

forma a tornar significativos os conceitos e relacioná-los com a prática, principalmente ao considerar-se a aprendizagem significativa em adultos. Quando o sujeito extrapola estes conhecimentos, ou seja, é capaz de implementar a sua prática em outras linguagens de programação pode ser considerado como letrado em código (LCod).

A partir do entendimento deste modelo, levando-se como base a aprendizagem significativa e a andragogia, foi proposto o curso denominado “Eu Programo 1.0!”, com duração de 20 horas. Para este, foram desenvolvidos materiais instrucionais, organizados em documentos (tutoriais, códigos fonte, entre outros) para serem utilizados nas aulas bem como instrumentos para avaliação. Para a prática deste foi eleito o software *App Inventor*⁷. A escolha do ambiente de programação foi porque, a partir de uma interface simples e intuitiva, o usuário é capaz de desenvolver seus aplicativos (App) em pouco tempo e com esforços reduzidos de programação. Ou seja, já nos primeiros encontros, o aluno consegue desenvolver um aplicativo simples. Lembrando que o App é um software desenvolvido para dispositivos móveis, logo, a avaliação (Habilidade 5) pode ser executada no próprio dispositivo do aluno.

Os primeiros App desenvolvidos em aula, tem como função formar subsunçores com relação ao uso da ferramenta e, portanto, não utilizaram nenhum conceito sugerido no MRPC, pois foram construídos para que o usuário entenda o funcionamento da ferramenta. Para os demais buscaram-se temas que fossem da realidade do aluno e dos seus interesses, pois conforme Knowles *et al.* “os adultos ficam prontos para aprender as coisas que têm de saber e para as quais precisam se tornar capazes de realizar a fim de enfrentar as situações da vida real” (2011, p. 74). Também, na escolha dos temas a serem desenvolvidos, foram avaliados aspectos como motivação do aprendiz, como, por exemplo, melhores oportunidades de empregos, salários mais altos e satisfação pessoal. Em síntese, quanto à escolha dos temas dos aplicativos, o modelo andragógico de Knowles foi um dos principais referenciais, assim 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, o envia para o professor, o qual faz a correção e retorna um *feedback*, porque “sabemos que a andragogia deve ser mais personalizada para se adequar à singularidade entre os adultos”(Knowles *et al.*, 2011, p. 146), assim o foco da aprendizagem está no aluno e cabe ao professor/tutor acompanhar este processo.

Os documentos construídos foram:

- Plano de aula: protocolo que contém todas as informações sobre o curso, tais como objetivo, tema, atividade práticas e materiais utilizados;

- Guia de aula: cada App tem o seu guia de construção, que serve para orientar o aluno adulto e autodirigido a construir seu aplicativo com ou sua orientação do professor.

- Contrato de Aprendizagem: Este documento serve para organizar e orientar sobre os objetivos do processo de ensino e aprendizagem, para todos os envolvidos neste, fomentando a autonomia do aprendiz (Knowles *et al.*, 2011).

- Testes: forma desenvolvidos pré-testes, testes intermediários e pós-testes. Os dois primeiros foram baseados no Bebras. O último teste foi projetado para avaliar todas as habilidades definidas no MRPC. Para cada questão proposta, foram assinaladas as habilidades correspondentes que estavam sendo avaliadas.

5.3. Instrumentos de Avaliação

Para iniciar a abordagem sobre a avaliação, retoma-se o objetivo deste trabalho que é de compreender e analisar o desenvolvimento do PC nos sujeitos adultos.

Há a recomendação de que muitos projetos sejam desenvolvidos ao longo do processo, a fim de oportunizar ao professor o entendimento do pensamento de cada aluno, além de serem programados vários pontos de verificação da aprendizagem durante os encontros, conforme sugerem Brennan e Resnick (2012). O uso de procedimentos diversos para obter os dados é imprescindível para aumentar a qualidade dos resultados encontrados (Gil, 2002).

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

⁷ Ferramenta de criação de App, disponibilizada gratuitamente na internet pelo MIT, em <http://ai2.appinventor.mit.edu/>.

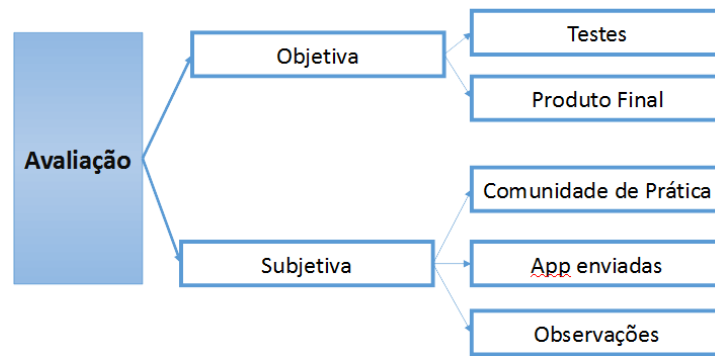


Figura 2 - Tipos de Instrumentos Avaliativos
 Fonte: autores (2019)

Os (instrumentos) objetivos estão compostos por testes (pré, pós e intermediários) e pelo produto final, com seus respectivos indicadores. Os instrumentos subjetivos são: observação, comunidade de prática (CoP) e aplicativos desenvolvidos durante o curso. Esta divisão serve para o

entendimento da proposta avaliativa, de forma estática. Já na Figura 3; **Error! No se encuentra el origen de la referencia.**, o processo de avaliação, através de um fluxograma, é mostrado.

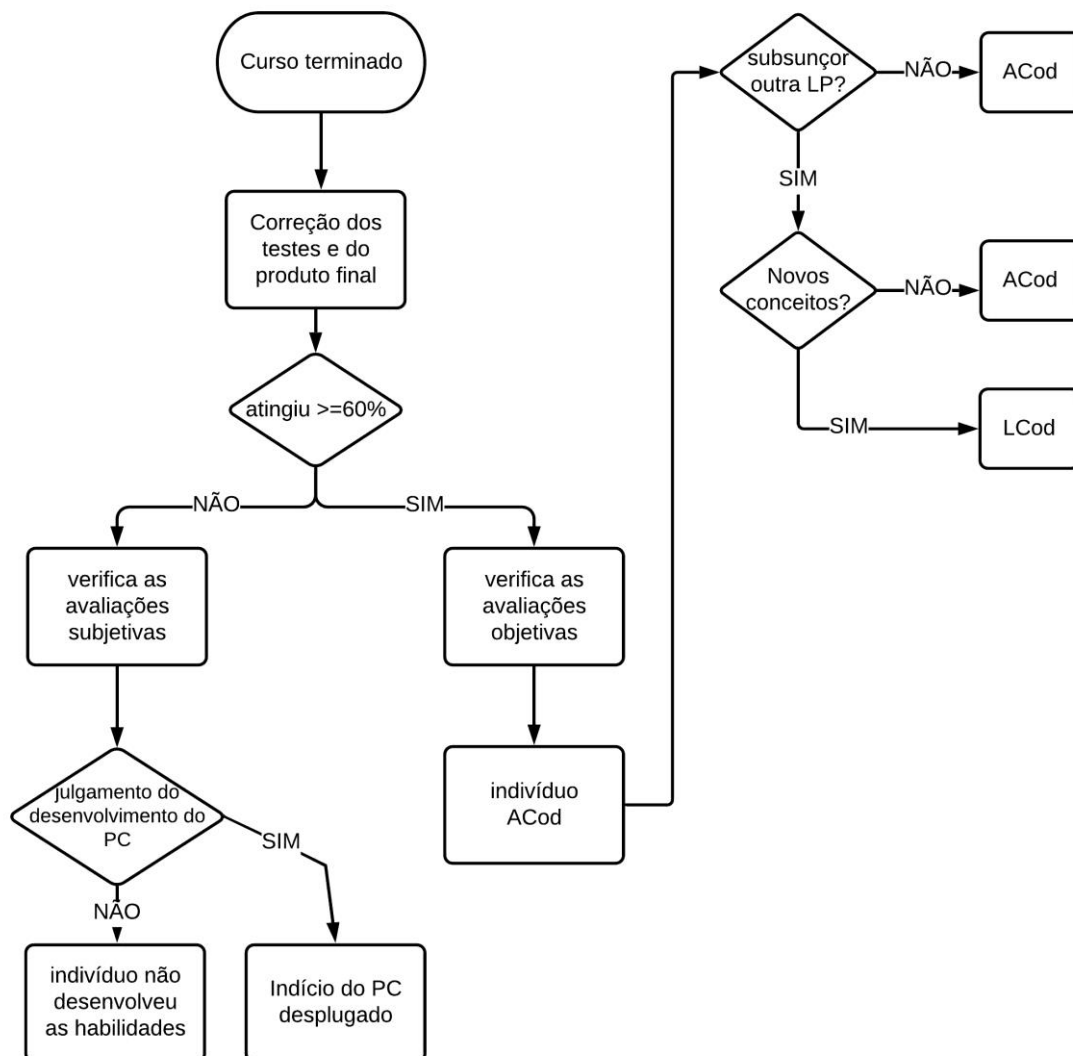


Figura 3 - Processo de avaliação para o MRPC

Fonte: Autores (2020).

Após todos os testes realizados e o produto final (App) entregue e devidamente avaliado, faz-se o somatório das questões dos testes e o resultado deverá atingir o valor mínimo de 60% de acertos. Este valor também deve ser observado para os indicadores do produto final. Com estes valores conhecidos, pode-se sugerir que o indivíduo está ACod. Já para os subjetivos, o professor deverá avaliar de forma individual cada aluno e, então, poderá determinar se o aluno desenvolveu todas habilidades propostas no MRPC, juntamente com as observações realizadas durante todo o curso e na CoP. Assim sendo, o professor poderá sugerir que o aluno desenvolveu o PC, porém de forma desplugada, ou seja, a habilidade 4 não foi desenvolvida. A seguir, será mostrado o detalhamento das avaliações.

5.3.1. Avaliação Objetiva

São propostos o pré-teste, um teste intermediário e o último chamado de pós-teste. As questões que compõem os dois primeiros testes foram retiradas do Bebras e, portanto, já validadas. Todas as questões dos testes possuem a alternativa “não entendi a questão”, tendo como fim avaliar a Habilidade de compreensão do sujeito. Sendo assim, evitando de errar a questão por falta de entendimento marcando uma resposta aleatória.

Para cada questão já fica registrado a qual(is) habilidade(s) ela está atrelada. Por exemplo, observe a Figura 4.

A International Society for Technology in Education (ISTE) enfatiza que o pensamento computacional na educação não está trazendo alunos para a posição de líder em Ciência da Computação, mas aplicando suas habilidades de pensamento computacional também em outros cursos como um hábito, aumentando a capacidade do indivíduo em resolver problemas do seu cotidiano.

Por isso, vamos resolver as questões!! Essas questões foram retiradas do teste Bebras. Estão disponíveis para responder no AVA (<http://ecompartindo.santana.ifsul.edu.br/>) ou envie sua resposta no WhatsApp (privado!! Com seu nome e as respostas).

A cor azul significa que o texto está escrito em espanhol e em verde na língua portuguesa. Se você não entendeu a questão e não sabe como resolvê-la, marque a alternativa NÃO ENTENDI A QUESTÃO.

2. Garrafas/ Botella

Un castor pone cinco botellas en una mesa. Las coloca para que cada botella se vea un poco. Pone la primera en la parte de atrás de la mesa y coloca cada botella subsecuente delante de la anterior.

Um castor coloca cinco garrafas em uma mesa. Ele as coloca de modo que cada garrafa apareça um pedaço. Ele coloca a primeira garrafa na parte de trás da mesa e cada garrafa subsequente na frente da anterior.



¿En qué orden fueron colocadas las botellas según la imagen?
Em que ordem as garrafas foram colocadas conforme mostrado na figura?

a) E D C B A
b) D B C A E
c) E C D A B
d) D C E B A
e) não entendi a questão / no entendí la pregunta

País proponente: Inglaterra
Habilidade: resolução de problema, abstração

Figura 4 - Exemplo da questão 2 do pré-teste
Fonte: autores (2019).


Na figura anterior, pode-se perceber que as questões são apresentadas em dois idiomas, o país que desenvolveu a questão e quais as habilidades envolvidas. Nos testes, as alternativas são de múltipla escolha, por isso são considerados como instrumento objetivo.

Sobre a avaliação do produto, denomina-se de Produto Final a definição e construção de um App. O tema é definido conjuntamente de acordo com os interesses e necessidades do grupo. Para avaliar o resultado do App proposto no curso, buscou-se a norma ISO/IEC 25010:2011 (*System and Software Quality Models*) que apresenta características e subcaracterísticas de qualidade de produto de software. Portanto, os critérios de qualidade foram cuidadosamente selecionados, depois as questões (indicador) da presença ou

não da subcaracterística avaliada foram descritos, no próximo passo um *template* para avaliação que poderá ser utilizado pelo tutor foi desenhado (Figura 5), bem como a escala da presença.

A conclusão desta avaliação, o professor deverá contar a presença dos indicadores, em um total de 10. Pelo menos 6 deverão estar presentes (60%), sendo obrigatórios todos os da característica Funcionalidade. Terminado este processo, ambas avaliações objetivas deverão ter atingido 60% para que o aluno seja considerado ACod.

A seguir, apresentam-se os indicadores e o modo de avaliar de forma subjetiva.



Avaliação do Projeto Final do Curso

Nome do APP:
 Nome do participante:
 Nome do Avaliador:
 Data:

Característica	Subcaracterística	Pergunta	Presença	
			Sim	Não
Aptidão Funcional	Completude funcional	Propõe-se a fazer o que é apropriado? Tudo o que foi planejado?		
	Correção funcional	Faz o que foi proposto de forma correta?		
	Adequação funcional	Propõe-se a fazer o que é apropriado?		
Usabilidade	Reconhecimento de adequabilidade	Atendeu suas necessidades?		
	Aprendizagem	É fácil de aprender?		
	Operabilidade	É fácil de operar e controlar?		
	Proteção contra erros do usuário	Tem validação de dados de entrada?		
	Estética da interface do usuário	A interface é agradável?		
Manutenibilidade	Testabilidade	É fácil de testar?		
Total				

Figura 5 - Avaliação do Produto Final
 Fonte: Adaptado pelos autores com base na ISO/IEC 25040.

5.3.2. Avaliação Subjetiva

São propostas três formas de avaliar o processo de desenvolvimento do PC, de maneira subjetiva, as quais são: a observação direta, o acompanhamento dos aplicativos desenvolvidos durante o curso e interação do aluno na CoP (Comunidade de Prática).

A observação direta é uma evidência que deve ser realizada pelo professor durante o transcorrer de suas aulas, ou seja, observar seus alunos, em tempo real e tratar de refletir e julgar o posicionamento do aluno de acordo com o contexto e o conteúdo sendo trabalhado. Sugere-se que o professor faça um diário de bordo com suas anotações, oriundas de sua observação e reflexão.

No decorrer das aulas, o aluno deve ser observado e, quando um aplicativo for finalizado, ou mesmo no final de cada

encontro, o professor solicita que o respectivo App seja disponibilizado, tanto para ele, quanto para o grupo de colegas. A partir do código-fonte, o professor poderá acompanhar a produção do aluno com relação ao software desenvolvido, verificar a completude, se houve erros, entre outros. Por conseguinte, deve fornecer um *feedback* adequado, pois, indubitavelmente, ao analisar o código fonte, pode-se perceber a presença (ou não) das habilidades referentes aquele App. Acredita-se que desta forma, o professor se torna capaz de traçar o perfil de conhecimento do seu aluno, de forma individualizada.

Ainda no viés da subjetividade, agregando a observação e a análise de desenvolvimento de cada aluno, cabe ao professor incentivar o uso da CoP, a fim de que o grupo compartilhe suas experiências (como já visto no acompanhamento dos App), suas dúvidas, seus acertos etc. A partir das interações dos alunos, encarrega-se o professor da observação destas comunicações para compor o perfil do aluno com relação ao desenvolvimento do seu PC, através das habilidades sugeridas no MRPC. Por exemplo, se um aluno apresenta na CoP uma dúvida e algum colega responde, então o professor já terá dados sobre as habilidades dos dois membros interagentes.

Ao considerar o sujeito como centro da aprendizagem, argumenta-se que o PC deve estar incorporado em todo o processo de ensinar, nos quais os professores devem propor casos para serem resolvidos baseados nas ideias de PC. Além disso, afirma que os alunos podem gradualmente entender o pensamento abstrato e o raciocínio lógico através de treinamento do pensamento e aplicação de métodos de PC na prática, pois o conhecimento do indivíduo está na habilidade de resolver problemas reais. Com isso, justifica-se a quantidade de atividades práticas sugeridas nesta pesquisa.

De um lado, ancorado em Ausubel (2003) e Knowles, Holton e Swanson (2011) e, de outro lado, no entendimento e avaliação de PC, propôs-se este constructo andragógico, que trata 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).

6. CONSIDERAÇÕES FINAIS

A ideia de resolução de problemas, algoritmos e programação pode ser remetida aos trabalhos de Papert, nos anos de 1980. Este autor definiu a linguagem LOGO, na qual as crianças programavam através do movimento de uma tartaruga, inspirado na teoria construtivista de Piaget. Mais tarde, Wing retoma esta proposta descrevendo-a como PC, porém de uma forma mais ampla, sem mencionar idade ou faixas etárias. De tal forma que muitas são as discussões e propostas acerca desta temática, não havendo, todavia, consenso da comunidade acadêmica sobre PC e como avaliá-lo.

Aliado a isto, há poucas publicações sobre este tema envolvendo adultos, se comparados às que tratam de crianças e adolescentes na educação básica. Portanto, de um lado, ancorado em Ausubel e Knowles e, de outro lado, no entendimento e avaliação de PC, a contribuição inédita e inovadora deste trabalho foi propor os constructos pedagógicos, que tratam tanto de desenvolvimento de habilidades para resolver problemas na forma de programas, ou não, quanto para avaliar este processo. Para atender ao objetivo deste trabalho foram propostos: um modelo de referência – MRPC, um curso para desenvolver o processo de ensino e aprendizagem dos sujeitos sobre PC e uma metodologia de avaliação. Dessa forma, estas proposições, tanto de ensino, quanto de aprendizagem que, aliadas, definem o sujeito ACod e avaliam o seu processo de construção de conhecimento sobre PC.

O MRPC lista habilidades básicas que o sujeito deverá desenvolver para pensar computacionalmente, além de propor a habilidade de Compreensão – inédita com referência aos autores pesquisados. Com respeito à Habilidade (resolução algorítmica), no *framework* proposto por Brennan e Resnick (2012), os autores sugerem conceitos e práticas a serem desenvolvidos, porém muitos conceitos estão definidos e a prática para este modelo utiliza o Scratch. Por isso, a seleção de conceitos básicos é mais uma importante contribuição desta pesquisa.

Como resultado, apresenta indícios das habilidades propostas, com a finalidade de afirmar que o sujeito tornou-se um ACod, ou desenvolveu as habilidades exceto a de programação (desplugado) ou ainda nenhuma das anteriores. Certamente cabe justificar que o processo proposto considera as recomendações da avaliação andragógica, a qual deve ser ininterrupta, diagnóstica e as falhas, tão logo se detectem, sejam rapidamente corrigidas.

Logo, desenvolver a habilidade de programação, atualmente se torna importante, segundo Ben Pring *et al.*, da Cognizant (empresa especializada em tecnologia)⁸, pois de cada 13 novas profissões que irão surgir, 10 exigirão conhecimento de linguagens de programação. Por isso, acredita-se que os currículos do Ensino superior deverão incorporar a disciplina de PC, podendo tomar como referência as propostas realizadas neste trabalho.

Uma importante contribuição está no método de avaliação, o qual entrelaça duas formas de avaliação, sendo estas complementares. Permite ao avaliador/tutor perceber nuances do processo de ensino e aprendizagem. Como por exemplo, a interação na comunidade de prática e sua participação durante as aulas. Também permite ao tutor o acompanhamento contínuo do processo de entendimento e programação dos App desenvolvidos.

⁸ Disponível em:
<https://www.cognizant.com/whitepapers/21-jobs-of-the-future-a-guide-to-getting-and-staying-employed-over-the-next-10-years-codex3049.pdf>. Acesso em 02 Abr 2019.

Retomando o já exposto, dois estudos de casos validaram este constructo e está em andamento um novo estudo (2021) e outros planejados até julho do próximo ano.

E para finalizar: 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. No Relatório da ONU *The Age of Digital Interdependence* (ONU, 2019), está discorrido sobre como será o gerenciamento das oportunidades e os riscos de uma rápida mudança tecnológica que impactará profundamente no futuro das pessoas e no futuro do planeta. E na junção desses dois temas, aliados com a situação das pessoas, especificamente no ano de 2020, atípico, pandêmico, essas questões se tornam realidade em um curtíssimo espaço de tempo. É preciso urgentemente buscar soluções.

Agradecimentos ao IFSUL pelo auxílio financeiro investido nesta pesquisa.

7. REFERENCIAS

- Araújo, A. L. S. O. de. (2019). *Quantifying Computational Thinking Abilities* [Tese (Doutorado em Ciência da Computação)]. Universidade Federal de Campina Grande.
- Ausubel, D. P. (2003). *Aquisição e retenção de conhecimentos: Uma perspectiva cognitiva. Tradução de Lígia Teopisto* (1st ed.). Plátano Edições Técnicas.
- Avila, C., Cavalheiro, S., Bordini, A., Marques, M., Cardoso, M., & Feijo, G. (2017). Metodologias de Avaliação do Pensamento Computacional: uma revisão sistemática. *Simpósio Brasileiro de Informática Na Educação (SBIE), XXVIII., 1, 113*. <https://doi.org/10.5753/cbie.sbie.2017.113>
- Brackmann, C. P. (2017). *Desenvolvimento do pensamento computacional através de atividades desplugadas na Educação Básica* [Tese (Doutorado em Informática na Educação), Universidade Federal do Rio Grando do Sul]. <https://doi.org/10.1111/j.1469-7610.2010.02280.x>
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *American Educational Research Association - AERA*, 1–25. https://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf
- Cavalcante, A., Costa, L. D. S., & Araujo, A. L. (2016). Um estudo de caso sobre competências do pensamento computacional desenvolvidas na programação em blocos no Code.Org. *Workshops Do Congresso Brasileiro de Informática Na Educação (CBIE), V., 1, 1117*. <https://doi.org/10.5753/cbie.wcbie.2016.1117>
- Cordenonzi, W. H. (2020). *O desenvolvimento do pensamento computacional e as evidências da alfabetização REIEC Año 16 Nro. 1 Mes Julio* 54
- em código em adultos [Tese (Doutorado em Ensino)]. Universidade do Vale do Taquari - Univates.
- CRTC. (2018). *Curriculo de Referência em Tecnologia e Computação*. <http://curriculo.cieb.net.br/>
- Dagiené, V., Sentance, S., & Stupuriené, G. (2017, March 30). Developing a two-dimensional categorization system for educational tasks in informatics. *Informatica, 28*(1), 23–44. <https://doi.org/10.15388/Informatica.2017.119>
- Gil, A. C. (2002). *Como elaborar projetos de Pesquisa* (4th ed.). Atlas.
- Gouws, L., Bradshaw, K., & Wentworth, P. (2013). First year student performance in a test for computational thinking. *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference on - SAICSIT, 13, 271*. <https://doi.org/10.1145/2513456.2513484>
- Grover, S., & Pea, R. (2013, January 1). Computational Thinking in K–12. *Educational Researcher, 42*(1), 38–43. <https://doi.org/10.3102/0013189X12463051>
- ISTE, & CSTA. (2011). *Operational Definition of Computational Thinking for K–12 Education*. <https://www.iste.org/explore/Solutions/Computational-thinking-for-all>
- 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. S., Holton, E. F., & Swanson, R. A. (2011). *Aprendizagem de Resultados* (2nd ed.). Elsevier.
- Koh, K. H., Basawapatna, A., Bennett, V., & Repenning, A. (2010). Towards the automatic recognition of computational thinking for adaptive visual language learning. *Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 59–66. <https://doi.org/10.1109/VLHCC.2010.17>
- Kordaki, M. (2010). A drawing and multi-representational computer environment for beginners' learning of programming using C: Design and pilot formative evaluation. *Computers and Education, 54*(1), 69–87. <https://doi.org/10.1016/j.compedu.2009.07.012>
- 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>
- Lye, S. Y., & Koh, J. H. L. (2014, December). Review on

teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61. <https://doi.org/10.1016/j.chb.2014.09.012>

Moreno León, J., Román González, M., & Robles, G. (2018). On computational thinking as a universal skill: A review of the latest research on this ability. *IEEE Global Engineering Education Conference (EDUCON)*, 2018-April(April), 1684–1689. <https://doi.org/10.1109/EDUCON.2018.8363437>

ONU. (2019). *The Age of Digital Interdependence - Report of the UN Secretary-General's High-level Panel on Digital Cooperation*. <https://digitalcooperation.org/wp-content/uploads/2019/06/DigitalCooperation-report-for-web.pdf>

Perrenoud, P. (1999). *Construir as competências desde a Escola* (1st ed.). Artmed.

Roazzi, A., & Hodges, L. (2013). Compreensão de texto e modelos teóricos explicativos: a influência de fatores linguísticos, cognitivos e metacognitivos. In A. Motta, Márcia P., Spinillo (Ed.), *Compreensão de texto*. Casa Psicólogo. c:%5CW A L no disco computador%5CDOUTORADO%5Caprendizagem e tipos

Román González, M. (2016). *Códigoalfabetización y Pensamiento Computacional en Educación Primaria y Secundaria: Validación de un Instrumento y Evaluación de Programas* [Tese (Doctorado En Educación)]. Universidad Nacional de Educación a Distancia.

Román González, M., Pérez González, J. C., & Jiménez Fernández, C. (2017, July). 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>

Seiter, L., & Foreman, B. (2013). Modeling the learning progressions of computational thinking of primary grade students. *Annual International ACM Conference on International Computing Education Research (ICER)*, 9., 59. <https://doi.org/10.1145/2493394.2493403>

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020, April 1). Assessing computational thinking: A systematic review of empirical studies. *Computers and Education*, 148, 103798. <https://doi.org/10.1016/j.compedu.2019.103798>

Van Dyne, M., & Braun, J. (2014). Effectiveness of a computational thinking (CS0) course on student analytical skills. *ACM Technical Symposium on Computer Science Education - (SIGCSE)*, 45., 133–138. <https://doi.org/10.1145/2538862.2538956>

Wiebe, E. N., London, J., Aksit, O., Mott, B. W., Boyer, K. E., & Lester, J. C. (2019). Development of a lean computational thinking abilities assessment for middle grades students. *ACM Technical Symposium on Computer Science Education (SIGCSE)*, 50., 2, 456–461. <https://doi.org/10.1145/3287324.3287390>

Wing, J. M. (2017, November 10). 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/>

Walkiria Helena Cordenonzi

Possui graduação em Informática pela Pontifícia Universidade Católica do Rio Grande do Sul; Especialização Em Ciências da Computação pela Universidade de Passo Fundo, Brasil; e mestrado em Ciência da Computação pela Universidade Federal do Rio Grande do Sul. Atualmente atua como docente do Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense, no curso binacional (Brasil e Uruguai), Câmpus Santana do Livramento. Tem experiência acadêmica na área de Educação, com ênfase em Ensino de Computação, atuando principalmente nos seguintes temas: lógica e programação e banco de dados e informática educativa. Atua em projetos de ensino, extensão e pesquisa e orientação de trabalhos de final de curso de graduação e pós-graduação.