

ENSINO DE PROGRAMAÇÃO E AS ESTRATÉGIAS PEDAGÓGICAS UTILIZADAS NO BRASIL

Roni Ferreira¹
Marco Braga²

RESUMO

O novo cenário mundial, político, econômico e social, exige a necessidade de se desenvolver práticas pedagógicas capazes de potencializar a apropriação do conhecimento de uma forma mais significativa e mais próxima da realidade. Entre estas práticas inovadoras, o Ensino de Programação tem ganhado cada vez mais espaço como promotora de uma ecologia cognitiva global. Neste contexto, o presente trabalho realizou uma análise de conteúdo nos artigos das principais revistas da área de Ensino, no período de 1996 a 2016, a fim de elucidar como os ambientes educacionais são transformados pela Programação de Computadores, proporcionando, assim, o protagonismo discente. Os resultados mostram uma conjunção de estratégias pedagógicas no âmbito brasileiro, que apontam para um “casamento” entre o Ensino da Programação e as Metodologias Ativas de aprendizagem (81%). Neste ambiente, o estudante está ativamente envolvido com o seu processo de ensino e aprendizagem, controlando e criando ritmos e novas relações com o conhecimento. Verificou-se, também, que os espaços criativos de Programação ampliam o entendimento sobre os impactos da ciência e tecnologia na sociedade, buscando por meio do aprender fazendo e da aprendizagem colaborativa, orientar resoluções de problemas reais e auxiliar na tomada de decisões, características de vidas imersas na Cibercultura.

Palavras-chave: Pensamento Computacional; Ensino de Programação; Metodologias Ativas; Ambientes Educacionais.

TEACHING OF PROGRAMMING AND PEDAGOGICAL STRATEGIES USED IN BRAZIL

ABSTRACT

The new world scenario, political, economic and social, demands the need to develop pedagogical practices capable of potentiating the appropriation of knowledge in a more meaningful way and closer to reality. Among these innovative practices, the Teaching of Programming has been gaining more and more space as a promoter of a global cognitive ecology. In this context, the present work performed a content analysis on articles from the main journals in the Teaching area, in the period from 1996 to 2016, in order to elucidate how educational environments are transformed by Computer Programming, thus providing student protagonism. The results show a conjunction of pedagogical strategies in the Brazilian scope, which point to a “marriage” between the Teaching of Programming and Active Learning Methodologies (81%). In this environment, the student is actively involved with his teaching and learning process, controlling and creating rhythms and new relations with knowledge. It was also verified that the creative spaces of Programming expand the understanding about the impacts of science and technology in society, seeking through learning by doing and collaborative learning, to guide real problem solving and assist in decision making, characteristics of lives immersed in cyberculture.

Keywords: Computational Thinking. Teaching Programming. Active Methodologies. Educational Environments.

Submetido em: 22/8/2020

Aceito em: 13/3/2021

¹ Autor correspondente: Instituto Federal de Educação, Ciência e Tecnologia do Rio de Janeiro. Pós-graduação em Engenharia de Produção e Sistemas. Av. Maria Luiza, s/n, Sacra Família do Tinguá. CEP 26660-000 - Engenheiro Paulo de Frontin/RJ, Brasil. <http://lattes.cnpq.br/1655236800124634>. <https://orcid.org/0000-0002-3695-6421>. profroniferreira@gmail.com

² Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Diretoria de Pesquisa e Pós-graduação - DIPPG. Rio de Janeiro/RJ, Brasil. <http://lattes.cnpq.br/5370405194554703>. <https://orcid.org/0000-0002-1289-9178>

INTRODUÇÃO

Quando uma tecnologia é substituída por outra, e assim sucessivamente, a escola é afogada por inovações e expectativas que não se concretizam, justamente por falta da realização de estudos e pesquisas aprofundados sobre a inclusão, consequências e resultados das Tecnologias Digitais de Informação e Comunicação (TDICs) nas salas de aula. Repensar as estratégias formais de ensino com o uso das TDICs, colocando o aluno como um agente ativo no seu processo de aprendizagem, pode diminuir a evasão escolar, o analfabetismo funcional e a desmotivação para estudar, que são mais suscetíveis ao vigente modelo de ensino tradicional e mecânico. Então, por que ainda perdemos tempo com aulas meramente expositivas, se elas não satisfazem as premissas de uma educação integrada com a realidade? A informação está hoje a um toque do *smartphone*, se tornando uma espécie de expansão da memória humana e tornando o ato de copiar do quadro uma tarefa sem sentido. Nóvoa (2019) nos lembra que as evidências de mudança social são claras, “mas a escola revela, sobretudo, uma grande incapacidade para pensar o futuro, um futuro que já faz parte da vida das nossas crianças” (p. 3).

A transmissão de informação deve ser trocada pela manipulação da informação, por meio de debates e discussões, explorações e simulações. Os ambientes de aprendizagem na Cibercultura precisam ser mais flexíveis, motivadores e dinâmicos. Todas estas mudanças influenciam métodos e conteúdos educacionais, pois “[...] a prioridade no ensino terá mudado da transmissão e organização de informações para o gerenciamento do conhecimento” (BATES, 2016, p. 129). Isto acontece devido ao momento histórico em que vivemos, cheio de imprevisibilidade e reordenamento das antigas estruturas sociais. Aplicar o conhecimento sabendo em qual contexto social se está trabalhando é de suma importância para uma educação significativa.

A necessidade de se compreender o avanço das TDICs no cenário educacional brasileiro se justifica pela necessidade de compreender quais são os impactos que este movimento provoca na redução das diferenças sociais existentes no país. Tal medição pode contribuir na elaboração de políticas públicas mais eficientes, além da atualização do conhecimento científico sobre o tema. Avaliações realizadas pelo Comitê Gestor da Internet no Brasil (CGI.br) procuram produzir indicadores da realidade brasileira sobre o acesso e o uso das TDICs na educação (CGI.BR, 2018). Segundo a pesquisa, houve mudanças na percepção dos professores sobre os impactos das TDICs nas práticas pedagógicas, provocando um aumento na adoção de novos métodos de ensino (85% em 2016 para 88% em 2017).

Portanto, é preciso que o docente introduza as tecnologias no seu dia a dia, criando relações imediatas com esta cultura digital. Ao fazer esta imersão voluntária, o docente entrará em um processo de aquisição de novas competências para ensinar no novo contexto que se apresenta (PERRENOUD, 2000). A relevância deste processo é justamente uma busca pela compreensão e entendimento de como os alunos aprendem. Caso não se alcance este entendimento, os professores continuarão realizando “casamentos” equivocados entre epistemologias de aprendizagem e tecnologias. A

escolha de ambas é dependente e elas se afetam mutuamente, tendendo ao fracasso ou ao sucesso na maneira como se ensina (BATES, 2016):

Ficará claro que a nossa escolha de abordagens de ensino, e até mesmo, o uso da tecnologia, é totalmente dependente de crenças e pressupostos que temos sobre a natureza do conhecimento, sobre os requisitos do assunto de nossa disciplina e sobre como pensamos que os alunos aprendem (BATES, 2016, p. 86).

Logo, é possível notar que independente da tecnologia que será usada, o quanto é nova ou inovadora, a escolha estratégica de como ela apoiará o ensino, terá impacto na forma como é introduzida na sala de aula e nos resultados de sua aplicação. Não é simplesmente ensinar sobre a tecnologia, é ensinar com a tecnologia, ensinar por meio da tecnologia, para aprofundar os saberes e engajar o aluno em práticas construtivas e criativas, assim:

[...] o elemento central não serão as aplicações e equipamentos, mas sim os conceitos que sustentam as tecnologias digitais, de modo a fornecer aos educandos autonomia para compreendê-las, utilizá-las e explorá-las mais profundamente (PAULA; VALENTE; BURN, 2014, p. 54).

Dentro desta perspectiva, os “Ambientes de Programação” já fazem parte do currículo escolar de vários países. Entendo que, neste trabalho, os “Ambientes de Programação” não se referem a nenhum tipo de interface específica ou *framework* para programar computadores, mas se refere aos espaços criados dentro da educação formal ou informal, em que os alunos aprendem conceitos e práticas do Pensamento Computacional para solucionar algum problema, de forma criativa e colaborativa. Neste ambiente, o conhecimento não é trabalhado de forma linear, mas abarca processos de ensino e aprendizagem mais dinâmicos e participativos (Ibid.). Seguindo as tendências globais, países, como o Brasil, buscam acompanhá-las e adaptá-las conforme as suas políticas para a área educacional e de acordo com as suas limitações locais, entre elas, a formação e qualificação profissional docente, pois:

Já não se trata de melhorias ou de aperfeiçoamentos, ou mesmo de inovações, mas de uma verdadeira metamorfose da escola. Fazer esta afirmação é, também, reconhecer as mudanças que, inevitavelmente, atingem os professores e a sua formação (NÓVOA, 2019, p. 3).

Neste sentido, a presente investigação busca compreender a seguinte questão: Quais são as estratégias pedagógicas utilizadas nos ambientes educacionais de Programação brasileiros, de forma a proporcionar uma aprendizagem ativa e contextualizada?

QUADRO TEÓRICO

Os pressupostos para a prática docente derivam de teorias de ensino e aprendizagem que foram construídas de acordo com o momento social e histórico em que surgem. O enfoque de cada uma pode variar de acordo com o conceito do ensinar e do aprender no qual se baseiam. Algumas tendem a focar mais no aluno, outras no professor ou nos procedimentos ou nos espaços educacionais. Fato é que, além destas referências teóricas, o professor também incorpora na sua prática docente muitas das

metodologias “de como foi educado durante sua vida escolar” (SANTOS, 2005, p. 31). Isto explica, um pouco, os motivos pelos quais muitas metodologias tradicionais ainda estejam enraizadas na forma como se ensina e na forma como se entende o processo de aprender. Dentro do atual contexto da Cibercultura, o enfoque em uma aprendizagem autônoma e flexível assume uma importância mais significativa para a participação social do que o ensino tradicional.

Estas novas metodologias são chamadas de “ativas”, pois invertem a centralidade dos processos de ensino e aprendizagem, colocando o aluno no seu âmago e dando-lhe liberdade na apropriação do conhecimento. Outro fator que o atual contexto social educativo tem assimilado é o uso das TDICs, que, por sua vez, provocaram novas propostas de compreensão da aprendizagem, como o Construcionismo de Papert, a Aprendizagem Significativa de Ausubel e o Conectivismo de Siemens (SCHELLER; VIALI; LAHM, 2014). Nas três teorias, a premissa principal é a motivação do aluno, contrariando o mecanicismo que se estabeleceu na educação durante séculos, pois para “aprender na era digital pressupõe um sujeito autônomo, conectivo, criativo, crítico, interativo e reflexivo perante as decisões que necessita tomar ao navegar pela rede” (Ibid., p. 9).

Então, a tecnologia se torna “como uma força disruptiva e uma oportunidade promissora” (CHRISTENSEN; HORN; JOHNSON, 2009, p. 17) para promover novos objetivos para os processos de ensino e de aprendizagem, ao mesmo tempo em que traz a necessidade de inovações metodológicas que maximizem a sua inserção no cenário educacional. No Ensino de Programação, por exemplo, algumas metodologias têm buscado mudar o senso comum de que o ato de construir um programa seja solitário e uma atividade isolada e de reclusão social (estereótipos encontrados nas mídias, como filmes e revistas). Pelo contrário, programar hoje se tornou uma tarefa extremamente colaborativa, devido à complexidade e à grandiosidade dos sistemas de informação que são disponibilizados no mercado. O programador contemporâneo necessita, antes de qualquer conhecimento técnico, saber ouvir, argumentar e expressar ideias. O diálogo com outros programadores, analistas de sistemas, analistas de banco de dados, especialistas em redes, entre outros profissionais, se torna uma essência vital para o bom desenvolvimento (ciclo de vida) de um sistema. Neste sentido, desenvolver a competência de colaboração, por exemplo, no ensino da Programação é uma estratégia compatível com as demandas de uma era digital (AQUINO FILHO; SCHIMIGUEL; AMARAL, 2016).

Na verdade, a Programação de Computadores tem sido vista como uma porta aberta para a popularização e divulgação do Pensamento Computacional nas escolas de Ensino Básico, pois engloba em suas raízes uma grande gama das competências e das técnicas, oriundas das Ciências da Computação (RAABE; RIBEIRO, 2017). Este “pensamento”, ou forma de ser e agir, promovem um desenvolvimento na dimensão cognitiva, como raciocínio lógico-matemático, organização e análise de dados, abstração e criação de modelos representativos da realidade, além da dimensão social, como o aprimoramento de habilidades comunicativas e colaborativas, também da dimensão afetiva, principalmente nos aspectos da perseverança e autoconfiança. Neste caso, a implementação de um “Ambiente de Programação”, isto é, um espaço de desenvolvimento do Pensamento Computacional permite não somente que os professores

trabalhem com seus alunos a resolução de problemas complexos, mas também, ao mesmo tempo, aprofunde a fluência digital discente para uma leitura e escrita de um mundo digital (VALENTE, 2019). Assim, o Pensamento Computacional, inserido por meio da Programação de Computadores, cria um espaço de interação e troca de conhecimento, onde a postura ativa e autônoma dos alunos, podem capacitá-los para serem os protagonistas das próximas interfaces digitais, influenciadores da cultura digital do futuro.

Desta forma, “Ambientes de Programação” são espaços inovadores de educação, onde se empreende uma busca por um novo fazer pedagógico, capaz de mudar práticas arbitrárias em práticas mais colaborativas e cooperativas, deixando o clima da sala de aula menos conflituoso, e passando a se configurar mais em um espaço de negociação de práticas e ideias. Pode-se dizer que se trata de uma “pedagogia da parceria” (PRENSKY, 2010, p. 41), que favorece o diálogo entre aluno-professor e aluno-aluno, ambos recebendo feedback da tecnologia, constroem juntos novas percepções de mundo. Neste tipo de ambiente, leva-se em consideração a interdisciplinaridade na aprendizagem dos alunos e os múltiplos contextos em que estão inseridos. O professor é responsável por delinear trilhas de aprendizagem, selecionar ou sugerir os recursos tecnológicos, além de planejar e preparar o layout do ambiente. Cada uma destas escolhas tem um papel relevante no despertar da autonomia discente (BRAGA; GUTTMANN, 2019). A partir disto, é possível perceber que, dentro destes ambientes de aprendizagem, o foco da avaliação está voltado mais para o processo de construção do conhecimento, ao invés de ser sobre o resultado das atividades. Logo, o *design* do ambiente, o que inclui as escolhas tecnológicas, os recursos didáticos disponíveis, as estratégias pedagógicas, os problemas a serem investigados e as diretrizes de condução dos projetos, interfere diretamente na gestão do conhecimento e acompanhamento das experiências dos alunos, isto é, do seu processo de aprender e do agir sobre e com o conhecimento.

As provocações dentro destes ambientes são realizadas para que o aluno procure refletir sobre o conhecimento que está sendo explorado, assumindo uma postura ativa de aprendizado individual e coletivo. Pensar sobre o que se está fazendo, durante o período de aula, e não depois, é um diferencial das metodologias ativas. Percebe-se que essa vertente metodológica une o pensar e o fazer, como os artesões pré-industriais faziam, pois “o artesão é o sujeito que, essencialmente, incorpora a união da atividade física com a atividade intelectual.” (CAMARGO, 2016, p. 68). Esta nova modalidade de aprendizagem experiencial se aproxima de uma tendência dos tempos atuais: a “Cultura Maker”, que tem provocado uma verdadeira revolução, em cadeias produtivas e em modelos educacionais. Este movimento, oriundo das tendências da Cibercultura, representa mais do que um pragmatismo educacional, reflete a “[...] condição humana de engajamento” (loc. cit.).

A partir do princípio de que o artesão pode existir na marcenaria, no laboratório científico ou em um conservatório de música, por que não no ambiente escolar?

Novas tecnologias, como o Arduino e Raspberry Pi, aliados à Programação, agregam maior capacidade de exploração de conteúdos e desenvolvimento de competências. Esta modelagem de problematização aproxima imaginário e realidade,

dando condições ao aluno de retomar conceitos e redesenhar caminhos mais significativos para ele, de forma que ao ser desafiado cognitivamente, aproveite as vantagens de uma aprendizagem personalizada e colaborativa (BRAGA; GUTTMANN, 2019). Portanto, dentro de um “Ambiente de Programação”, além da programação de robôs, pode-se construir games e aplicativos móveis, em que o conhecimento ganha um novo sentido, saindo de um aspecto meramente consumista e passando a ser matéria-prima de transformação.

METODOLOGIA DE PESQUISA

Destarte, para realizar a pesquisa em questão, foi escolhida a metodologia qualitativa da Análise de Conteúdo, seguindo a fundamentação conceitual e a prática sistematizada de Bardin (2011). A pré-análise constituiu-se da escolha de artigos que formaram o *corpus* da análise, levando em consideração o princípio da exaustividade. O recorte temporal foi de 20 anos, iniciando em 1996 até 2016. A pertinência dos artigos corresponde às revistas da área de Ensino que possuem repositórios on-line, classificadas pela CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), em 2016, como publicações de Qualis A1, A2, B1, B2 e B3.

Os critérios para manter o princípio da homogeneidade do *corpus*, formado pelos artigos encontrados, foi selecionar os artigos, em que a escola ou o docente construíram um “Ambiente de Programação”, isto é, um ambiente educacional de prática de aprendizagem de Programação, no qual havia participação discente e indícios no texto de quais foram as metodologias e as tecnologias usadas.

As palavras-chave usadas para extrair os artigos pertinentes aos critérios estabelecidos foram: Ensino de Programação, Ensino de Algoritmos, Ensino de Códigos, Aprendizagem de Programação, Linguagem de Programação, Programação de Computadores, Lógica de Programação, Programação Educacional, Codificação como Alfabetização, Fluência Digital, Scratch, Linguagem LOGO e Software LOGO.

A segunda fase, após reunir todo o material e prepará-lo para realizar uma exploração apurada e mais profunda dos textos, foi formada principalmente pela codificação e agregação de recortes textuais em unidades de registro (Ibid.). Estas unidades foram classificadas em categorias simbólicas (MORAN, 2015). Dentre as categorias, criou-se uma variável de categorização textual intitulada de “Tipo de Metodologia”, que poderia receber os seguintes valores:

- 1 para Metodologias Tradicionais;
- 2 para Metodologias Ativas;
- 3 para Metodologias Mistas (tradicionais e ativas juntas); e
- 4 para Metodologias Não Definidas, conforme Quadro 1:

Quadro 1 - Categorização das Estratégias Usadas para Ambientes de Programação

Tipo de Metodologia	Código	Descrição
<i>Metodologias Tradicionais</i>	m1	São baseadas na exposição de conteúdo, em que o professor mantém controle sobre as tarefas e permite que a participação do aluno seja meramente repetitiva e memorística.
<i>Metodologias Ativas</i>	m2	Constrói-se um ambiente desafiador para o aluno, despertando e permitindo sua criatividade e autonomia. São baseadas no desenvolvimento de competências, pois o importante não é acumular conteúdo, e sim, saber usá-lo ao longo da vida.
<i>Metodologias Mistas</i>	m3	Oferta-se uma tecnologia inovadora, mas se limita a criatividade e a autonomia do aluno, além de se continuar focando somente em conteúdo. Une-se tecnologia nova com uma pedagogia ultrapassada ou vice-versa.
<i>Metodologias Não Definidas</i>	m4	Não foi possível identificar a metodologia empregada, seja pela falta de clareza na construção textual ou pela análise de referências que não apontaram autores de linhas pedagógicas conhecidas.

Fonte: Elaborado pelos autores, 2020.

Foram pesquisadas um total inicial de 1382 revistas, sendo que após os refinamentos sucessivos e os procedimentos aplicados, chegou-se a um total de 703 títulos investigados. Por fim, a terceira fase foi constituída da análise dos 104 artigos, que preencheram os requisitos essenciais deste trabalho. Utilizou-se também, nesta fase, o software IRaMuTeQ (Interface R para Análise Multidimensional de Textos e Questionários), de origem francesa e de acesso livre (LOUBÈRE; RATINAUD, 2016), que além de realizar várias análises lexicais, incrementou maior riqueza nas possibilidades de interpretações e inferências.

RESULTADOS E ANÁLISE

Durante a pesquisa, encontrou-se uma conformidade, mesmo dentro deste escopo de mudanças, causado pelas TIDCs na educação, cenários onde as formas de implementação impedem a integração entre estratégia pedagógica e o recurso tecnológico escolhido. No Brasil, em alguns casos (19%), oferta-se a Programação de Computadores (PC) por meio de um viés metodológico tradicional (m1), isto é, deixa-se o aluno vislumbrar todas as possibilidades que ele pode criar com a ferramenta, só que ele fica impedido de fazê-lo. É como dar corda em um pião, e ainda assim, querer que ele fique parado. Quando o resultado é mais importante do que a trajetória de reflexão, de autocrítica, de erros e acertos, de empoderamento do conhecimento, torna-se evidente a preocupação com o “aprender a programar” em detrimento do “aprender programando” (RESNICK *et al.*, 2009). Destes 19%, mais da metade (55%) foi considerada uma espécie de híbrido, usando metodologias tradicionais e ativas ao mesmo tempo. Neste íterim, realizou-se uma análise de especificidades, utilizando as variáveis de caracterização *m_1 (tradicionais), *m_3 (mistas) e *m_4 (não identificadas). Na Figura 1, é possível visualizar a frequência das palavras nos artigos agrupados com as metodologias citadas e aquelas palavras que apresentam maior constância nos segmentos de textos.

e convicções (LOUBÈRE; RATINAUD, 2016). A comunicação escrita mostra, então, uma compreensão mais apurada do planejamento estratégico existente nas diferentes abordagens pedagógicas apresentadas pelos autores.

Observe que todas as nuvens, conforme Figura 1, não apresentam em sua centralidade a palavra “aluno”, ou “discente”, ou “aprendiz”. Em maiores detalhes, como dito anteriormente, na nuvem inferior-vermelha (metodologias tradicionais), nota-se que o professor ainda está, nestes casos, no controle absoluto do processo de aprendizagem de Programação. Apesar de a palavra *estudante* aparecer nos limites da nuvem, de igual maneira, as palavras “comando”, “linguagem computacional” e “resultado” estão no mesmo patamar de importância, dando ênfase às abordagens pedagógicas que priorizam formatos mais rígidos de aprendizagem. Alinhado a este pensamento, a nuvem superior verde, que representa os artigos com metodologias mistas, também não enfoca a participação do aluno e demonstra em seu núcleo a relevância demasiada que é dada ao conteúdo. As palavras “jogo” e “projeto” aparecem por serem as metodologias usadas, mas que fornecem ao aluno roteiros, ideias e ritmos pré-formatados, cabendo ao discente apenas segui-los em uma sequência inflexível e desfavorável à criatividade. Do lado esquerdo, na cor azul, encontram-se os artigos cujas metodologias não se conseguiram identificar. Curiosamente, destaca-se a palavra “valor”, e cuja concordância textual foi levantada com a ajuda de funções do software. Tal palavra está associada à parametrização, isto é, encontra-se em vários segmentos de texto os termos “valor de parâmetro”, “cálculo de valor”, “valor de função” ou “retorno de valor”.

Nestes artigos, o estudo em questão apontou para o uso limitado da Programação de Computadores como promotora dos princípios do Pensamento Computacional (VALENTE, 2019), isto é, as metodologias usadas fornecem ao aluno roteiros prontos, sem margem para agregação de ideias e com ritmos inflexíveis de aprendizagem. Dentro deste escopo engessado, o “Ambiente de Programação” coloca o aluno como um mero seguidor das orientações do professor e fica tolhido de utilizar autonomia e criatividade nas suas decisões e escolhas.

Diante deste quadro, observa-se que as metodologias tradicionais e mistas se diferenciam, principalmente, das metodologias ativas pelo fato de não considerarem competências em seus objetivos de aprendizagem. Em um mundo onde facilmente se obtém a informação em qualquer lugar e tempo, pois a memória humana está sendo armazenada em nuvem, tornou-se altamente relevante e necessário desenvolver competências para se trabalhar com muita informação disponível. A resposta certa não é mais suficiente, é preciso fomentar a reflexão daquilo que se aprende, justificar e dar novo sentido, pois “o conhecimento não é estático, mas cresce e se desenvolve nos alunos. Em particular, em uma era digital, significa desenvolver habilidades tanto quanto acumular conteúdo” (BATES, 2016, p. 580). Na verdade, muitos pesquisadores e autores da área de Ensino e Tecnologias Educacionais apontam para uma necessidade de se trabalhar os conteúdos de forma interdisciplinar, quando se opta por uma abordagem ativa (MORAN, 2015; BATES, 2016).

Learning. Este aspecto corrobora com um alinhamento do Ensino de Programação Brasileiro com estratégias aplicadas em países que já possuem um pouco mais de experiência na área (FERREIRA; DUARTE, 2019). Orbitando a palavra “aluno”, na Figura 2, observa-se que o papel do professor não perde sua importância neste processo educacional, ao contrário, reafirma sua expertise em apontar caminhos a serem trilhados pelos estudantes, que podem concordar ou discordar, argumentar e justificar suas opiniões. O Quadro 2 mostra as metodologias ativas:

Quadro 2 – Metodologias Ativas nos Ambientes de Programação no Brasil

Metodologia (m2)	PARTICIPAÇÃO CORPUS TEXTUAL	Nível Educativo	Fase HS
<i>Aprendizagem Baseada em Problemas</i>	29%	I, B, S, FP e EE	1, 2 e 3
<i>Gamificação</i>	18%	B, FP e EE	3
<i>Aprendizagem por Projetos</i>	13%	B e S	3
<i>Aprendizagem Híbrida</i>	8%	B e S	3
<i>Robótica Pedagógica</i>	6%	B e S	3
<i>CSCL e Comunidades de Prática</i>	4%	B e S	2 e 3
<i>Investigação-Ação Educacional (IAE) e a Educação Dialógica-Problematizadora (EDP)</i>	4%	B e S	3
<i>Aprendizagem Experiencial</i>	2%	M e S	2 e 3
<i>CTS e História da Ciência</i>	2%	M	3
<i>Metodologias Ágeis</i>	2%	S	3
<i>Personalized System of Instruction (PSI)</i>	1%	B e S	3
<i>Technological Pedagogical Content Knowledge (TPACK)</i>	1%	FP	3
<i>Taxonomia de Bloom</i>	1%	B e S	3

Nível Educativo: I = Infantil; B = Básica; M = Médio; S = Superior; FP = Formação de Professores; EE = Educação Especial.

Fonte: Elaborado pelos autores, 2020.

De acordo com o Quadro 2, o professor agora se torna um maestro do “*design* de caminhos” educacionais (MORAN, 2015, p. 26), tendo a sua disposição uma diversidade de estratégias pedagógicas pautadas em uma postura ativa do aluno, como as adotadas nos ambientes inovadores de aprendizagem de Programação do Brasil. Ainda no mesmo quadro, o total de artigos onde não foi possível definir as metodologias empregadas é igual a 9% (m4). Observa-se, também, que cada metodologia usada pelos docentes brasileiros contém o nível educacional correspondente e uma fase histórico-social (Fase HS), na qual elas foram empregadas, o que facilita entender mudanças temporais de abordagens ativas nos “Ambientes de Programação”. Segundo Ferreira e Duarte (2019), a fase 1 vai desde 1996 até 2000, tendo como características o Programa Nacional de Tecnologia Educacional (PROINFO) – programa governamental criado em 1997 para inserir tecnologias nas escolas públicas do país – e a linguagem LOGO. A fase 2 inicia-se

em 2001 até 2007, com destaque para sistemas com base na Internet e priorizando o raciocínio lógico-matemático. A fase 3 (2008 até 2016) tem como diferenciais a preocupação com o perfil do aluno e o surgimento de interfaces mais amigáveis, como o *Scratch*, para o aprendizado de Programação. A metodologia de Aprendizagem Baseada em Problemas ou *Problem-Based Learning* (PBL) é uma das abordagens pedagógicas mais usadas pelos professores de Programação (29%), estando presente em todas as fases históricas e níveis de ensino encontrados na pesquisa. Os aspectos de interação professor-aluno e aluno-aluno são frequentes nos debates e discussões que buscam resoluções para o problema e nas observações dos trabalhos de colegas. As aulas são geralmente realizadas em laboratórios de informática para experimentação de fenômenos naturais, situações da realidade e para criar ambientes de simulação.

Outros trabalhos optaram por aderir à Aprendizagem por Projetos (13%) e pela Abordagem Experiencial (2%). Ambas, valorizam o engajamento dos alunos pelo princípio “*learning by doing*”, buscando fornecer uma aprendizagem ativa de Programação, muito mais abrangente para o aluno, com a incorporação de práticas e costumes da Cultura *Maker*. A criação destes ambientes desafiadores aproxima o estudante iniciante das estratégias e estruturas mentais usadas pelo Pensamento Computacional (VALENTE, 2019). Os artigos apontam que estas estratégias desenvolvem principalmente a responsabilidade, a autonomia e a criatividade. Os alunos precisam organizar as fases do projeto, tomar decisões, reconhecer padrões, fazer a modularização do problema, acompanhar e controlar o fluxo de dados e mapear camadas distintas da aplicação (dados, código e interface) desenvolvidas por eles (ROBINS; ROUNTREE; ROUNTREE, 2003).

A aprendizagem colaborativa (4%), representada pela abordagem de Comunidades de Prática e pela *Computer Supported Collaborative Learning* (CSCL) ou Aprendizagem Colaborativa Assistida por Computador, integra o grupo de metodologias ativas nos “Ambientes de Programação” brasileiros. Estas estratégias mencionadas apoiam a comunicação interpessoal e dinamizam a cultura do compartilhamento no contexto escolar. As atividades propostas requerem, além de participação ativa, constante negociação entre pensamento e ação, situações da realidade e modelos abstratos, linguagens e significados, interesses individuais e coletivos. As construções e reconstruções das histórias narradas em forma de código, pelos alunos, acontecem não só de forma interativa com a máquina, mas também com os colegas. Evidencia-se explicitamente o objetivo de encorajar os alunos a vivenciarem a cultura da sociedade em rede (BATES, 2016; CHRISTENSEN; HORN; JOHNSON, 2009; AQUINO FILHO; SCHIMIGUEL; AMARAL, 2009; BRAGA; GUTTMANN, 2019).

Os projetos organizacionais da contemporaneidade estão imersos em condições de imprevisibilidade e de adaptabilidade às constantes mudanças. Portanto, a pesquisa demonstra que, em alguns casos, elas são transportadas pelos professores para a área educacional. As Metodologias Ágeis, por exemplo, utilizadas na Programação empresarial, buscam, neste novo cenário, desenvolver a coragem dos alunos para lidarem com situações adversas, além de aumentar o empenho na aplicação de práticas incrementais e iterativas e o respeito pela contribuição dos colegas (CRESPO; GÄRTNER; GÓMEZ, 2012). Todo este arcabouço tecno-metodológico consegue contribuir para

que o discente de diferentes níveis educacionais, conforme os dados levantados, tenha uma postura ativa frente aos novos tempos. Neste contexto, os professores brasileiros, nos trabalhos selecionados, estão experimentando novas técnicas para tornar a Programação atraente aos seus alunos. Entre estas iniciativas, a Gamificação ganhou destaque (18%) para fomentar o ensino de Programação. Uma das características desta metodologia é a promoção do encontro entre aprendizado e diversão, personificando o que Papert dizia sobre o processo de criação da Programação, que era “*hard fun*”, isto é, exige dedicação e esforço, mas ao mesmo tempo, é prazeroso e divertido (RESNICK, *et al.* 2009). Neste sentido, os artigos que trazem esta dinâmica dos jogos para o ensino de Programação, como oportunidade de construção de ambientes de aprendizagem desafiadores, requerem criatividade, concentração, (re)avaliação de ideias e estratégias, (auto)correção de erros e oferta de feedback constante.

Percebe-se que a relevância da contextualização social e histórica tem aumentado nas estratégias docentes, que envolvem o ensino científico e tecnológico. Dentre elas, a Investigação-Ação Educacional (IAE) e a Educação Dialógica-Problematizadora (EDP), em que os alunos trabalham em pares ou grupos as temáticas estudadas e, a cada passo, criam seminários de debate para reflexão do conteúdo investigado. Trata-se de metodologias no qual o diálogo com a realidade e com os outros contribui para aumentar o senso crítico individual (BASTOS; MULLER; ABEGG, 2013). A Programação é utilizada de forma ativa, principalmente para simular situações que confrontam visões prévias dos alunos, com visões científicas e tecnológicas, passando pelas fases de planejamento, ação, observação e reflexão, podendo retornar à primeira durante todo o processo. Assim, constatou-se com a pesquisa, que não se tem buscado apenas o aprimoramento da técnica, mas também, do pensamento crítico, levando debates de temas socialmente pertinentes sobre os impactos de C&T na vida dos alunos e da sociedade.

A *Technological Pedagogical Content Knowledge* (TPACK) foi uma abordagem pouco explorada pelos professores-autores (1%), apesar de ser utilizado como metodologia na formação de professores, que posteriormente aplicarão a Programação em suas salas de aula. Destacam-se também a PBL e a Gamificação como alternativas que estão sendo trabalhadas na formação docente. Dentro do escopo brasileiro, a presença do movimento Ciência, Tecnologia e Sociedade (CTS) e da Aprendizagem Experiencial como alternativas para o ensino de Programação vem ganhando espaço de forma marcante no Ensino Médio (RAABE; RIBEIRO, 2017; FERREIRA; DUARTE, 2019), justamente quando o jovem começa a desenvolver competências de empregabilidade, em que o pensar, o senso crítico e a reflexão precisam estar unidos ao fazer, ao agir e ao produzir coletivamente. Isto demonstra a versatilidade do Ensino de Programação de produzir técnicas que se moldam facilmente a processos pedagógicos interdisciplinares, sendo capaz de gerar um ambiente onde se consegue trabalhar a matemática, ciência e tecnologia, e ao mesmo tempo, ética, sustentabilidade e história. Justamente, este diálogo entre saberes é o que enriquece o Pensamento Computacional, como uma nova linguagem de raciocínio, de relacionamento, de agir e interagir, caminhando em consonância com o perfil de uma escola como lugar de uma “construção comum”, pois

Hoje em dia, a fragmentação a que estamos a assistir no ciber mundo, coloca a escola perante a urgência de valorizar a nossa pertença a uma mesma humanidade e a um mesmo planeta. Este *comum* não vem de *comunidade de identidade*, mas sim de *comunidade de trabalho*, isto é, o que fazemos em comum uns com os outros independentemente das nossas origens, crenças ou ideias (NÓVOA, 2020, p. 4-5).

Desta forma, a escola, que agrega o digital e a diversidade, passará a ser um espaço de articulação com outras instituições da sociedade para a construção de cidades inteligentes, que sejam realmente mais inclusivas, sustentáveis e transformadoras.

A investigação trouxe outros pontos interessantes para a área, como a percepção de que a maioria das metodologias ativas, citadas no trabalho, começaram a se desenvolver após 2007, quando o ensino de Programação começou a tomar uma proporção maior no cenário brasileiro (FERREIRA; DUARTE, 2019). Os resultados, até então apresentados, confirmam que as metodologias ativas (m2) condicionam os “Ambientes de Programação” dentro de uma perspectiva ativa de aprendizagem, em que o processo de ensino-aprendizagem possui centralidade no protagonismo discente, pois oferece maior plasticidade e adaptabilidade à aprendizagem de conteúdos e ao desenvolvimento de competências, necessárias para vidas imersas na Cibercultura.

CONSIDERAÇÕES FINAIS

A pesquisa de forma contundente revela uma conjunção das estratégias pedagógicas no escopo brasileiro, que apontam para um “casamento” entre o Ensino de Programação e as Metodologias Ativas (81%) ou aprendizagem centrada no aluno, como é referenciada mundialmente. Ambas as práticas se têm influenciado mutuamente e construído ambientes de aprendizagem promissores para avanços mais significativos na área educacional do Brasil. Além das percepções dos pesquisadores, recorreu-se as representações gráficas geradas pelo IRaMuTeQ, para realçar os resultados das técnicas de Análise de Conteúdo, aplicadas nos artigos selecionados da área de Ensino e dentro de um recorte temporal de 20 anos (1996 até 2016). Isto possibilitou visualizar a frequência das palavras nos artigos agrupados com as metodologias pedagógicas citadas e aquelas palavras que apresentaram maior constância nos segmentos de textos. Ressalta-se que a proporção de tamanho das palavras indica sua relevância, participação e frequência na linguagem textual dos artigos. Logo, as nuvens geradas são uma interpretação gráfica, que demonstram uma mudança drástica entre os modelos tradicionais e os modelos ativos de ensino e aprendizagem. A centralidade do ALUNO, na nuvem de palavras onde a Programação cria um Ambiente de Aprendizagem Ativa, revela, por meio da comunicação escrita, uma compreensão mais apurada do planejamento estratégico existente nas diferentes abordagens pedagógicas, empregadas pelos professores-autores.

O professor não perde sua importância, pelo contrário, se posiciona agora, como um facilitador e mediador, dando ênfase ao *design* do ambiente de aprendizagem e a orientação de trilhas flexíveis de conteúdos e competências. Respondendo à indagação inicial, apresenta-se as estratégias pedagógicas, que aparecem com maior frequência nos trabalhos analisados de Ensino de Programação no país: a PBL (29%), a Gamificação (18%), as Metodologias Tradicionais (19%), a Aprendizagem por Projetos (13%), a

Aprendizagem Híbrida (8%), com suporte presencial e on-line, e a Robótica Pedagógica (6%). Com exceção das Metodologias Tradicionais, que mantêm uma rigidez de seus roteiros descontextualizados, as demais formas de ensino promovem a democratização e a flexibilização do processo de ensino e aprendizagem. Estas Metodologias Ativas contribuem para que a Programação agregue os saberes de forma interdisciplinar, a diversidade de participação e expressão, de colaboração e representação das visões de mundo dos alunos.

Sendo assim, evidencia-se com a apuração do presente trabalho investigativo, que os ambientes inovadores de aprendizagem de Programação promovem o protagonismo discente no Brasil, das seguintes formas:

- a) os professores constroem processos de ensino e de aprendizagem mais flexíveis e mais centrados no aluno;
- b) buscam aproximar as realidades da Era Digital do cotidiano da sala de aula;
- c) aumentam a interatividade por meio da colaboração;
- d) mesclam os saberes de diferentes áreas de conhecimento;
- e) valorizam mais o processo do que o resultado; e
- f) facilitam a simbiose entre os conceitos abstratos e as atividades práticas.

Dentro desta perspectiva mais ampliada de ensino e aprendizagem científica e tecnológica e que corresponde aos anseios, práticas e resultados apurados na maioria dos artigos que fazem parte do *corpus* da pesquisa, detectou-se que o conhecimento adquirido não se limita apenas para armazenar conteúdo, antes, se apresenta como potencial de transformação da realidade, orientando escolhas e favorecendo projetos de vida dos discentes.

REFERÊNCIAS

- AQUINO FILHO, G.; SCHIMIGUEL, J.; AMARAL, L. Ambientes colaborativos para Ensino de Eletrônica e Lógica de Programação. *Revista de Ensino de Ciências e Matemática – RENCiMa*, v. 7, n. 5, p. 31-39, dez. 2016.
- BARDIN, L. *Análise de Conteúdo*. 4.ed. Lisboa: Edições 70, 2011.
- BASTOS, F.; MULLER, F.; ABEGG, I. Educação Científico-Tecnológica de Jovens e Adultos Mediada por Tecnologias Livres. *Revista Contexto & Educação*, v. 21, n. 76, p. 151–174, 2013.
- BATES, A. W. *Educar na era digital: design, ensino e aprendizagem* (Coleção tecnologia educacional). São Paulo: Artesanato Educacional, 2016.
- BRAGA, M.; GUTTMANN, G. The Knowledge Networks in a Makerspace: The Topologies of Collaboration. *International Journal of Science and Mathematics Education*, v. 17, n. 1, p. 13-30, fev. 2019.
- CAMARGO, C. O artesão, o filósofo e o maker. *Revista Digital de Tecnologias Cognitivas – TECCOGS*, n. 13, p. 67-77, jun. 2016.
- CGI.BR. *Pesquisa sobre o uso das Tecnologias de Informação e Comunicação nas Escolas Brasileiras - TIC Educação 2018*. Comitê Gestor da Internet no Brasil. Disponível em: <http://data.cetic.br/cetic/explore?idPesquisa=TIC_EDU>. Acesso em: 02 set. 2018.
- CHRISTENSEN, C.; HORN, M.; JOHNSON, C. *Inovação na sala de aula: como a inovação disruptiva muda a forma de aprender*. Porto Alegre: Bookman, 2009.
- CRESPO, S.; GÄRTNER, V.; GÓMEZ, A. Facilitando o aprendizado de programação sob a ótica de metodologias ágeis. *Revista Brasileira de Informática na Educação*, v. 20, n. 1, p. 54-68, ago. 2012.
- FERREIRA, R.; DUARTE, S. Ensino de programação: trajetória histórico-social e os avanços na cultura digital do Brasil. *Revista Brasileira de Ensino de Ciência e Tecnologia - RBECT*, v. 12, n. 1, p. 407-429, abr. 2019.

- LOUBÈRE, L.; RATINAUD, P. *Documentation IRaMuTeQ 0.6 alpha 3 version 0.1*. 2016, França. Disponível em: <<http://www.iramuteq.org>>. Acesso em: 28 nov. 2018.
- MORAN, J. Mudando a educação com metodologias ativas. In: SOUZA, C.; MORALES, O. (Orgs.). *Coleção Mídias Contemporâneas. Convergências Midiáticas, Educação e Cidadania: aproximações jovens. Vol. II*. PG: Foca Foto-PROEX/UEPG, 2015. p. 15-33.
- NÓVOA, A. Os Professores e a sua Formação num Tempo de Metamorfose da Escola. *Educação & Realidade*, Porto Alegre, v. 44, n. 3, e84910, 2019.
- PAULA, B.; VALENTE, J.; BURN, A. O uso de jogos digitais para o desenvolvimento do currículo para a educação computacional na Inglaterra. *Currículo sem Fronteiras*, v. 14, n. 3, p. 46-71, dez. 2014.
- PERRENOUD, P. *Dez novas competências para Ensinar*. Tradução Patrícia Chittoni Ramos. Porto Alegre: Artmed, 2000.
- PRENSKY, M. *Teaching Digital Natives: partnering for real learning*. California: Corwin, 2010.
- RAABE, A.; RIBEIRO, L. (Coords.). *Referenciais de Formação em Computação: Educação Básica – Versão julho/2017*. Disponível em: <<https://www.sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf>>. Acesso em: 23 mar. 2018.
- RESNICK, M.; MALONEY, J.; MONROY-HERNÁNDEZ, A.; RUSK, N.; EASTMOND, E.; BRENNAN, K.; MILLNER, A.; ROSENBAUM, E.; SILVER, J.; SILVERMAN B.; KAFAI, Y. Scratch: Programming for Everyone. *Communications of the ACM*, v. 52, n. 11, nov. 2009.
- ROBINS, A.; ROUNTREE, J.; ROUNTREE, N. Learning and Teaching Programming: a review and discussion. *Computer Science Education*, v. 13, n. 2, p. 137–172, jun. 2003.
- SANTOS, R. V. Abordagens do processo de ensino e aprendizagem. *Integração*, v. 11, n. 40, p. 19-31, maio 2005.
- SHELLER, M.; VIALI, L.; LAHM, R. A aprendizagem no contexto das tecnologias: uma reflexão para os dias atuais. *Revista Novas Tecnologias na Educação - RENOTE*, v. 12, n. 2, dez. 2014.
- VALENTE, J. A. Pensamento Computacional, Letramento Computacional ou Competência Digital? Novos desafios da educação. *Revista Educação e Cultura Contemporânea*, v. 16, n. 43, p. 147-168, 2019.

Todo conteúdo da Revista Contexto & Educação está
sob Licença Creative Commons CC – By 4.0