

A importância da implementação progressiva de ferramentas lúdicas no ensino-aprendizagem de programação

The importance of the progressive implementation of playful tools in programming education

Peterson Gomes de Moura Barros¹, Larissa de Oliveira Figueira², Ana Paula Rodrigues³, Francisco Conti Bauke⁴

¹Facti – Fundação de apoio à Capacitação em Tecnologia da Informação – Campinas – SP – Brasil

{peterson.barros,larissa.oliveira,ana.rodrigues,francisco.bauke}@facti.com.br

Resumo. Este artigo discute a eficácia de uma sequência didática progressiva no ensino de programação, composta por três etapas: linguagens baseadas em ícones, em blocos e, por fim, programação aplicada com microcontrolador. Fundamentado na Pedagogia das Competências e em abordagens cognitivas consolidadas, o estudo propõe essa progressão como alternativa às metodologias tradicionais que iniciam com linguagens textuais, frequentemente gerando barreiras de aprendizagem. A pesquisa adota uma metodologia quali-quantitativa, combinando análise documental e dados empíricos do Projeto Letramento Digital. Os resultados evidenciam maior coesão do percurso entre componentes curriculares, com atenuação de rupturas sobretudo em cenários de implementação estabilizada. Em contextos ainda em transição, observam-se efeitos parciais e sensíveis a fatores operacionais. As evidências sugerem que a progressão organiza e sustenta a transição conceitual quando acompanhada de tempo de prática e mediação pedagógica, contribuindo para trajetórias formativas mais coerentes.

Palavras-chave: implementação progressiva, pensamento computacional, lógica de programação, progressão didática, linguagens visuais.

Abstract. This paper examines the effectiveness of a progressive instructional sequence for introductory programming structured in three stages: icon-based environments, block-based languages, and application-oriented textual programming with a microcontroller. Grounded in the Pedagogy of Competencies and cognitive frameworks, the proposal is advanced as an alternative to traditional pathways that start directly with textual syntax and often impose early learning barriers. Using a mixed-methods design, we combine documentary analysis with empirical data from the Letramento Digital Project. Results indicate greater curricular cohesion across com-

ponents and attenuation of discontinuities, particularly where implementation conditions are stable. In cohorts still undergoing transition, effects are partial and sensitive to operational factors such as delivery format, infrastructure, and instructional continuity. The evidence suggests that the progression organizes and supports the conceptual shift toward textual programming when accompanied by protected practice time and sustained pedagogical mediation, contributing to more coherent learning trajectories and informing inclusive programming curricula.

Keywords: Gradual implementation, computational thinking, programming logic, didactic progression, visual languages.

1. Introdução

A programação computacional tem se consolidado como uma habilidade transversal de relevância crescente no contexto educacional. Esse cenário exige que os processos formativos se adaptem, incorporando metodologias pedagógicas que favoreçam o desenvolvimento integrado de competências cognitivas, técnicas e socioemocionais. No entanto, observa-se que a abordagem tradicional, pautada na introdução direta às linguagens de programação estruturadas e baseadas em texto, impõe desafios substanciais aos estudantes em fase inicial de aprendizagem, independentemente da faixa etária. Ambientes em blocos reduzem barreiras iniciais porque a gramática visual evita combinações inválidas e minimiza erros de sintaxe, permitindo que iniciantes concentrem-se no raciocínio e na resolução de problemas (Resnick et al., 2009). Em convergência, Kelleher e Pausch (2005) sistematizam estratégias para tornar linguagens mais acessíveis, simplificar a linguagem, restringir o domínio e prevenir erros de sintaxe, justamente para que o esforço cognitivo não fique preso à forma textual. Revisões recentes reforçam que linguagens em blocos favorecem o reconhecimento sobre a recordação, reduzem a carga cognitiva e previnem erros, contribuindo para a transição posterior ao código textual (Bau et al., 2017).

Este estudo é fundamentado na análise do ensino de programação direcionado a jovens e adultos de diferentes áreas de formação, abrangendo tanto aqueles que já tiveram contato prévio com a programação quanto aqueles que se aproximaram do tema pela primeira vez durante o percurso formativo. A heterogeneidade dos perfis dos participantes, tanto em relação à faixa etária quanto à experiência anterior, constitui um elemento central para a compreensão dos desafios pedagógicos enfrentados, exigindo a adoção de estratégias didáticas inclusivas e progressivas que favoreçam a aprendizagem significativa para estudantes com trajetórias acadêmicas e profissionais diversas.

Nesse contexto, emerge a importância da utilização progressiva de ferramentas lúdicas, estruturadas a partir das linguagens baseadas inicialmente em ícones, passando por linguagens em blocos até alcançar a programação textual. De acordo com Wing (2006), uma sequência didática progressiva pode favorecer a mobilização de competências cognitivas e o pensamento computacional; evidências de que percursos que iniciam em ambientes visuais e evoluem para linguagens textuais apoiam a transição entre níveis de

abstração são discutidas por Noone e Mooney (2018) e por Robins, Rountree e Rountree (2003).

Para sustentar essa abordagem, este estudo se fundamenta na Pedagogia das Competências. Para sustentar essa abordagem, este estudo se ancora na Pedagogia das Competências. À luz de Perrenoud (2000), a aprendizagem é entendida como um processo integrado no qual o estudante mobiliza e articula habilidades, atitudes e saberes para enfrentar situações-problema, reais ou simuladas. Nessa perspectiva, o ensino de programação deixa de ser compreendido como a simples apropriação de uma linguagem específica, sendo entendido como um processo contínuo de aquisição e integração de competências cognitivas, lógicas e criativas que capacitam o estudante a agir com autonomia e eficácia diante de diferentes desafios.

A partir das contribuições teóricas de Piaget e Vygotsky, reforça-se a importância de respeitar o desenvolvimento cognitivo e o papel essencial da mediação pedagógica para uma aprendizagem eficaz. Para Piaget (1976), o conhecimento deve ser construído progressivamente, respeitando os estágios evolutivos do pensamento, que partem de ações concretas para a abstração simbólica. Já Vygotsky (1991) destaca o papel central da interação social e da mediação realizada por ferramentas culturais e tecnológicas na internalização do conhecimento.

Além disso, estudos como o de Oliveira (2013) evidenciam que o ensino inicial de programação com linguagens textuais estruturadas pode resultar em desmotivação e abandono, especialmente quando os estudantes não conseguem compreender a lógica algorítmica por trás das instruções sintáticas. O autor aponta que a introdução abrupta ao código textual frequentemente limita a construção do raciocínio lógico, pois o foco inicial tende a recair sobre a memorização de comandos e regras gramaticais da linguagem, em detrimento da compreensão conceitual.

Este estudo examina essa progressão com base em dados administrativos e evidências de desempenho, empregando uma abordagem quali-quantitativa e um delineamento comparativo “antes-depois” entre turmas com sequência progressiva e turmas em formato tradicional. Para dar suporte metodológico às inferências, (i) operacionalizamos os desfechos de conclusão, aprovação e evasão; e (ii) aplicamos estatística descritiva (contagens, percentuais e diferenças em pontos percentuais) com exclusão de desistentes do denominador.

Foram estabelecidas como questões de pesquisa:

- RQ1. Em que medida a progressão atenua a queda de desempenho entre componentes curriculares consecutivos?
- RQ2. Quais condições contextuais (perfil dos participantes, infraestrutura, organização didática) potenciam ou limitam os efeitos observados e que implicações pedagógicas decorrem para o desenho de trilhas formativas?

O artigo contribui ao: (i) testar empiricamente o efeito da progressão em públicos adultos e heterogêneos; (ii) integrar matrizes teóricas, tornando visíveis tensões e limites na educação tecnológica.

Essa proposta de capacitação é realizada por meio do Projeto Letramento Digital, uma iniciativa da Fundação de Apoio à Capacitação em Tecnologia da Informação (FACTI), desenvolvida em parceria com o Ministério da Ciência, Tecnologia e Inovação (MCTI). O projeto tem como objetivo oferecer a capacitação integral dos participantes, com ênfase na apropriação crítica, significativa e criativa das tecnologias digitais, no desenvolvimento de competências digitais, cognitivas e socioemocionais, e na promoção da equidade no acesso ao conhecimento.

Embora o presente estudo apresente evidências favoráveis à proposta de progressão didática, é importante destacar algumas limitações metodológicas. Este trabalho não contemplou de forma aprofundada as possíveis interferências externas que podem ter influenciado os resultados obtidos, tais como o perfil socioeconômico dos estudantes, a infraestrutura das instituições participantes ou o grau de familiaridade prévia dos estudantes. Esses fatores, embora relevantes, não foram objeto de controle neste estudo e devem ser considerados em investigações futuras que busquem aprofundar a análise de variáveis contextuais.

Para sustentar essa proposição, apresenta-se uma fundamentação teórica interdisciplinar, discutem-se resultados de estudos de caso e propõe-se uma análise do impacto da adoção da sequência didática progressiva no Projeto Letramento Digital.

2. Fundamentação teórica

A abordagem por competências tem se consolidado como uma diretriz central nas políticas e práticas educacionais contemporâneas, especialmente em contextos que envolvem o uso das tecnologias digitais e o ensino de programação. A Pedagogia das Competências, conforme delineada por Perrenoud (1999) e aprofundada por Zabala e Arnau (2010), propõe uma educação que transcenda a mera transmissão de conteúdos e promova a mobilização integrada de conhecimentos, habilidades e atitudes em situações concretas e complexas. Ensinar por competências, nesse sentido, é favorecer a construção de sujeitos capazes de enfrentar problemas reais com criatividade, pensamento crítico e autonomia. Inserida nesse modelo pedagógico, a programação assume um papel central, destacando-se como elemento estruturante para o desenvolvimento do raciocínio lógico, da capacidade de resolução de problemas, da comunicação por meio de símbolos e da produção criativa orientada à inovação.

Do ponto de vista da aprendizagem significativa, essa perspectiva pressupõe que o conhecimento adquirido pelos estudantes seja funcional, ou seja, aplicável em diferentes contextos e socialmente relevante. Segundo Ausubel (2003), a aprendizagem torna-se significativa quando a nova informação se ancora de modo não arbitrário e substantivo ao conhecimento prévio do aprendiz, favorecendo a internalização de conteúdos com sentido para sua realidade. Nesse sentido, a simples memorização de comandos ou a repetição mecânica de instruções mostra-se incompatível com os objetivos formativos defendidos pela Pedagogia das Competências. Para Perrenoud (2000), ensinar por competências implica promover a mobilização de recursos cogniti-

vos, afetivos e sociais em situações complexas e autênticas, superando uma lógica escolar baseada apenas na transmissão de conteúdos. É, portanto, necessário construir trilhas formativas que favoreçam a autonomia intelectual e a capacidade de transferir saberes para novas situações, articulando teoria e prática em processos cada vez mais reflexivos, críticos e criativos (Zabala; Arnau, 2010).

Esse percurso formativo exige uma progressão didática intencional e fundamentada em teorias psicológicas do desenvolvimento e da aprendizagem. Com base na epistemologia genética de Jean Piaget (1975), compreende-se que os sujeitos percorrem estágios cognitivos sucessivos, do sensório-motor ao operatório formal, cada qual marcado por diferentes níveis de abstração e operações mentais. Crianças em fase operatória concreta, por exemplo, aprendem com maior eficácia quando envolvidas em atividades que exigem manipulação concreta e representação visual, o que torna inadequada a introdução imediata de linguagens textuais estruturadas, que exigem raciocínio simbólico abstrato e domínio metacognitivo.

Nesse aspecto, destaca-se a relevância da mediação simbólica e cultural proposta por Lev Vygotsky (1991), cuja teoria histórico-cultural compreende o desenvolvimento humano como um processo socialmente constituído. O conceito de Zona de Desenvolvimento Proximal (ZDP) salienta a importância das ferramentas culturais como mediadoras entre o sujeito e o conhecimento. Linguagens visuais de programação, como aquelas baseadas em ícones ou blocos, funcionam como instrumentos que ampliam a capacidade de aprendizagem ao facilitar a internalização de conceitos computacionais por meio da interação com pares e educadores.

Embora a literatura inicial sobre o ensino de programação tenha se concentrado amplamente no público jovem, pesquisas recentes sustentam que a abordagem sequencial de ensino é altamente eficaz também para adultos, especialmente aqueles que ingressam na área com níveis variados de familiaridade prévia. Robins, Rountree e Rountree (2003) argumentam que barreiras cognitivas no ensino de programação inicial são comuns em todas as faixas etárias, e que o uso imediato de linguagens textuais pode aumentar as dificuldades de abstração e a sobrecarga cognitiva. Luxton-Reilly (2016) reforça que a aprendizagem de programação sequencial, partindo de ambientes visuais e evoluindo gradualmente para linguagens textuais, oferece um caminho mais acessível e sustentável para adultos iniciantes, permitindo-lhes internalizar conceitos complexos de forma progressiva e significativa.

Além disso, revisões e estudos indicam que abordagens visuais e lúdicas são pertinentes em cursos profissionalizantes e processos de requalificação, favorecendo ambientes mais inclusivos e engajadores (Dahalan; Alias; Shaharom, 2024). Meta-análises em educação em programação apontam efeitos positivos especialmente sobre motivação e também sobre desempenho acadêmico (Zhan et al., 2022), enquanto revisões de escopo em Computação descrevem ganhos de engajamento e aprendizagem duradoura (Videnovik et al., 2023). Em relação à ansiedade, a evidência é heterogênea: há estudos que não observam redução significativa, embora se verifiquem menores cargas cognitivas e recomendações de design pedagógico para mitigá-la (Unal; Topu, 2021; Pan; Harun, 2025).

Corroborando essa perspectiva, uma análise da literatura, aponta que os impactos do ensino tradicional de programação textual em estudantes iniciantes indicam que, além da baixa retenção de conceitos, esse modelo pode afetar negativamente a motivação e o engajamento, criando barreiras cognitivas significativas. Defende-se, portanto, a adoção de metodologias mais acessíveis e progressivas, que respeitem os ritmos e as trajetórias individuais dos aprendizes, especialmente no contexto da formação inicial em computação.

Considerando tais fundamentos, práticas educacionais que promovem o ensino de programação de forma gradual têm ganhado destaque. Essas iniciativas educacionais de capacitação vêm sendo desenvolvidas com base nesses princípios, utilizando um conjunto de ferramentas tecnológicas articuladas em uma sequência progressiva. Na fase inicial de aprendizagem, são empregadas linguagens baseadas em ícones, como o ScratchJr, iniciativa voltada a estudantes que terão o primeiro contato com a programação e caracterizada pela manipulação intuitiva de elementos gráficos que representam comandos. Essa fase permite o desenvolvimento de noções de sequência, causalidade e estruturação de ideias, promovendo um letramento digital introdutório.

Em uma segunda etapa, adota-se o Scratch, uma linguagem baseada em blocos visuais que permite a criação de algoritmos mais sofisticados por meio de estruturas condicionais, repetições e variáveis. Desenvolvido no MIT Media Lab por Mitchel Resnick, o Scratch incorpora os princípios do construcionismo, teoria proposta por Papert (1980), segundo a qual o conhecimento é construído ativamente quando o aprendiz está engajado na criação de artefatos significativos e nos princípios da aprendizagem criativa. Por meio da programação de jogos, animações e histórias interativas, os estudantes ampliam suas capacidades de abstração, planejamento e expressão, ao mesmo tempo em que desenvolvem a colaboração e o pensamento computacional. Como argumenta Resnick (2009), programar torna-se uma forma de pensar, criar e comunicar.

Por fim, na terceira etapa do percurso, são utilizadas ferramentas que fazem a transição entre a programação em blocos e a textual, como o MakeCode, especialmente quando associado ao uso de dispositivos físicos, como o Micro:bit. Essa plataforma híbrida permite ao estudante visualizar simultaneamente os códigos em blocos e sua tradução em linguagem textual, articulando os conhecimentos provenientes de ambos os campos. Ao programar sensores, LEDs e atuadores físicos, favorece-se a transposição do pensamento teórico para a resolução de problemas práticos, o que fortalece o vínculo entre teoria e prática, reforçando a motivação, a compreensão e o senso de autoria.

A progressão aqui descrita não é resultado de uma escolha técnica ou de conveniência metodológica, mas sim de uma opção pedagógica embasada em evidências teóricas e empíricas. Estudos como o de Lima e Menezes (2024) apontam que, nas disciplinas introdutórias, as principais dificuldades concentram-se na resolução de problemas, na estruturação lógica e na compreensão da sintaxe de linguagens de programação. Esses fatores se relacionam, na literatura, a maior carga cognitiva quando o ensino começa di-

retamente em linguagens textuais, o que pode afetar o engajamento inicial, evitando que a ausência de uma mediação progressiva e acessível comprometa o processo de aprendizagem e torna o ensino ineficaz para a maioria dos estudantes.

Nos últimos anos, o ensino de programação com ferramentas visuais tem sido objeto de diversas investigações que destacam sua eficácia no desenvolvimento do pensamento computacional e na motivação dos estudantes. Sínteses recentes sobre ferramentas de feedback/correção automatizada em cursos introdutórios destacam ganhos na prontidão do retorno e no acompanhamento formativo, com potencial para reduzir frustração em tarefas de programação (Messer et al., 2023). Além disso, Costa e Kologeski (2024) realizaram uma revisão sistemática da literatura sobre o ensino de programação, identificando que métodos quantitativos são predominantes nas pesquisas sobre educação em programação, com ênfase na utilização de questionários e testes de desempenho como instrumentos de coleta de dados. Esses estudos reforçam a importância de metodologias baseadas em evidências para a implementação eficaz de práticas pedagógicas no ensino de programação.

No contexto da educação básica, Casagrande e Freitas (2023) destacam que a introdução de ferramentas visuais de programação, como linguagens baseadas em blocos, facilita a compreensão de conceitos abstratos e estimula o interesse dos estudantes pela área de computação. Essas abordagens são particularmente eficazes quando integradas a estratégias de ensino que consideram as características cognitivas dos alunos, promovendo uma aprendizagem mais significativa e contextualizada.

Nesse contexto, tornam-se fundamentais os referenciais teóricos e metodológicos que orientam projetos de capacitação para o Letramento Digital sustentados pela Pedagogia das Competências como princípio estruturante. Iniciativas que organizam o ensino de programação por meio de ferramentas progressivas configuram-se como importantes recursos pedagógicos, oferecendo subsídios para a construção de trilhas formativas coerentes com o desenvolvimento humano integral, a promoção da inclusão digital e a formação de sujeitos críticos, criativos e autônomos. O presente estudo apoia-se nesses fundamentos para investigar, à luz da teoria e da prática, os efeitos e as potencialidades de uma proposta formativa em programação alinhada às competências essenciais à participação ativa e reflexiva na vida contemporânea.

Embora a abordagem Pedagógica escolhida subsidie a progressão didática aqui analisada, reconhecem-se limites e tensões: risco de tecnicismo na operacionalização de competências; dificuldade avaliativa em captar simultaneamente fluência sintática e competências transversais; e dependência de condições de implementação (tempo de prática, mediação, infraestrutura). A literatura também registra divergências quanto ao timing da transição para

a linguagem textual, especialmente em contextos com heterogeneidade de perfis, o que explica parte da variabilidade empírica observada neste estudo.

3. Abordagem Metodológica

Considerando a proposta pedagógica do Projeto Letramento Digital e as evidências iniciais que indicam impactos positivos da adoção progressiva de ferramentas lúdicas no processo de ensino-aprendizagem de programação. Delineou-se o presente estudo com base em duas questões centrais: (RQ1) em que medida a progressão didática atenua a queda de desempenho entre componentes curriculares consecutivos (PC \square FD \square RP); (RQ2) quais condições contextuais potenciam ou limitam tais efeitos e que implicações decorrem para o desenho de trilhas formativas.

3.1 Objetivo Geral

A pesquisa aqui descrita teve por objetivo geral avaliar a eficácia da sequência didática progressiva de ferramentas lúdicas no ensino de programação, em comparação à abordagem tradicional, verificando seu impacto no desempenho e na aprendizagem dos estudantes. Essa abordagem é investigada como estratégia para o desenvolvimento de competências, especialmente aquelas relacionadas ao raciocínio lógico e ao pensamento computacional, considerando as particularidades cognitivas e os desafios enfrentados por estudantes com diferentes níveis de familiaridade prévia com a programação.

3.2 Objetivo Específico

Para alcançar o objetivo proposto, este estudo estabelece como objetivos específicos: analisar se a introdução dessa progressão contribuiu para o nivelamento da curva de aprendizagem entre os diferentes cursos que compõem o Módulo Essencial; e examinar a evolução dos resultados nas turmas sucessivas, considerando os efeitos da implementação gradual da nova metodologia ao longo do tempo.

Para alcançar os objetivos propostos, esta pesquisa adota uma abordagem metodológica mista, composta por elementos qualitativos de natureza exploratória e documental, complementados por uma análise quantitativa baseada em estudos de caso. A combinação dessas abordagens visa articular a compreensão teórica da progressão didática no ensino de programação com a análise empírica de seus impactos no processo de aprendizagem dos estudantes. A investigação qualitativa fundamenta-se na análise de documentos institucionais e referenciais teóricos, enquanto a vertente quantitativa apoia-se em dados coletados no âmbito de experiências educativas reais, com foco na avaliação de desempenho e engajamento dos participantes. Essa configuração metodológica é especialmente relevante, considerando-se o público heterogêneo estudado, permitindo compreender como a progres-

são didática influencia a aprendizagem de estudantes com trajetórias acadêmicas e profissionais distintas.

3.3 Abordagem Qualitativa

O procedimento metodológico incluiu a análise documental institucional do Projeto Letramento Digital, que constitui o principal objeto empírico deste estudo. A análise concentrou-se na estrutura metodológica do projeto, nos impactos observados entre os participantes e na aderência aos pressupostos teóricos da Pedagogia das Competências, com o objetivo de verificar a aplicação prática da sequência didática progressiva proposta nesta pesquisa.

Além disso, o estudo foi complementado por uma revisão sistematizada da literatura, baseada em trabalhos publicados em periódicos acadêmicos indexados nas bases Scielo, Periódicos CAPES, ERIC e Google Scholar. A busca foi orientada pelos descritores: ensino de programação, linguagens visuais, progressão didática, lógica computacional e pedagogia das competências.

Como critérios de inclusão, foram selecionados exclusivamente estudos que:

- apresentassem dados quantitativos claros e verificáveis;
- fossem conduzidos em contextos educacionais reais;
- abordassem o uso de linguagens baseadas em ícones, blocos ou linguagem textual no contexto de aprendizagem progressiva em programação.

3.4 Abordagem Quantitativa

Para a análise quantitativa, este estudo utilizou dados de 568 estudantes ao todo, participantes do Módulo Essencial do Projeto Letramento Digital nas cidades de Pato Branco e Londrina. Esse módulo abrange cinco cursos, dos quais três deles são cursos introdutórios de programação (Pensamento Computacional, Fundamentos de Programação e Robótica e Prototipagem), oferecidos sequencialmente. Em Pato Branco, foram acompanhadas 4 turmas (coortes denominadas T1 a T4), com 122, 40, 85 e 31 ingressantes em cada turma, respectivamente. Em Londrina, participaram 5 turmas (T1 a T5), com 34, 31, 79, 93 e 53 ingressantes, respectivamente. Assim, a cidade de Pato Branco contribuiu com 278 alunos e Londrina com 290 alunos na base de dados analisada. Os participantes são jovens e adultos de perfis diversos, incluindo professores da rede pública municipal e membros da comunidade sem formação prévia em computação, refletindo a heterogeneidade mencionada no contexto do projeto.

É importante destacar que, por desenho do projeto, diferentes turmas seguiram metodologias distintas. Nas turmas iniciais de cada cidade (até T1 em Pato Branco e T1–T3 em Londrina), adotou-se a metodologia tradicional de ensino: o curso de Pensamento Computacional era realizado com atividades desplugadas (sem o uso de linguagens de programação digitais) e, em seguida, os alunos eram introduzidos diretamente a uma linguagem de programação textual (Portugol) no curso de Fundamentos de Programação. Nas turmas posteriores (a partir da T2 em Pato Branco e da T4 em Londrina),

implementou-se a sequência didática progressiva de ferramentas lúdicas: o curso de Pensamento Computacional passou a incluir introdução à lógica de programação por meio de ferramentas visuais (linguagem em ícones), o curso de Fundamentos de Programação utilizou programação em blocos (plataforma Scratch), e somente no curso de Robótica e Prototipagem introduziu-se os primeiros conceitos da programação textual integrada a projetos práticos e a programação em Blocos (MakeCode com micro:bit). Essa distinção permitiu formar dois grupos de análise, turmas com abordagem tradicional vs. turmas com abordagem progressiva, preservando relativa comparabilidade, já que os conteúdos e critérios de avaliação dos cursos permaneceram alinhados ao currículo do projeto em ambas as cidades.

Os instrumentos de coleta de dados consistiram nos registros administrativos oficiais do Projeto Letramento Digital, contendo as informações de matrícula e desempenho de cada aluno em cada curso do módulo. Para cada estudante, obtiveram-se as notas finais e a situação de aprovação (Aprovado) ou não aprovação, incluindo casos de reprovação por nota e de desistência (Abandono) em cada disciplina. Esses registros foram extraídos das planilhas institucionais de histórico dos alunos e validados pela equipe do projeto. Não foram aplicados testes ou instrumentos adicionais de avaliação além dos já existentes no contexto dos cursos, de modo que a análise baseia-se em dados reais de desempenho acadêmico dos participantes.

Como indicador de aprendizagem, optou-se por utilizar principalmente a taxa de sucesso em cada curso e no módulo como um todo. Assim, a proporção de alunos aprovados/reprovados fornece uma medida objetiva do aprendizado consolidado ao término de cada etapa. Embora notas numéricas médias tenham sido consideradas como informação complementar e a conclusão bem-sucedida de cada disciplina, foi adotada como o indicador sumário de desempenho educacional. Esse indicador é válido no contexto do projeto pois os conteúdos e avaliações foram elaborados alinhados à Pedagogia das Competências, de forma que a conclusão com aproveitamento reflete a aquisição das habilidades e conhecimentos esperados em cada fase.

Além disso, por serem dados provenientes de avaliações oficiais do curso, eliminam-se vieses de auto-relato e garantem-se critérios uniformes de medida de aprendizagem para todos os participantes. Para cada componente e turma, foram considerados apenas Aprovados e Reprovados (Desistentes excluídos). O indicador principal é $\% \text{Aprovação} = A/(A+R)$. A comparação pré-pós define-se por (PB: T1 vs. T2-T4; LD: T1-T3 vs. T4-T5).

4. Resultados e análises

A análise dos resultados fundamenta-se em dados quantitativos obtidos no âmbito do Projeto Letramento Digital, executado pela Fundação de Apoio à Capacitação em Tecnologia da Informação (FACTI) nos municípios de Pato Branco e Londrina, entre 2022 e 2024. Todas as taxas reportadas consideram apenas aprovados e reprovados, com desistentes excluídos do denominador. O indicador principal é $\% \text{ de aprovação} = \text{Aprovados}/(\text{Aprovados}+\text{Reprovados})$. As contagens de concluintes (n) são apresentadas apenas como volume de corte e não como medida de desempenho.

Os estudantes participantes devem ser aprovados em todos os cinco cursos, Criatividade e Inovação, Pensamento Computacional, Estratégias Pedagógicas, Fundamentos de Programação e Robótica e Prototipagem, que compõem o módulo, sendo esperado que o número de concluintes se mantenha constante ao longo das diferentes etapas, sem ocorrências de evasão entre os cursos. Essa sequência contempla três etapas articuladas:

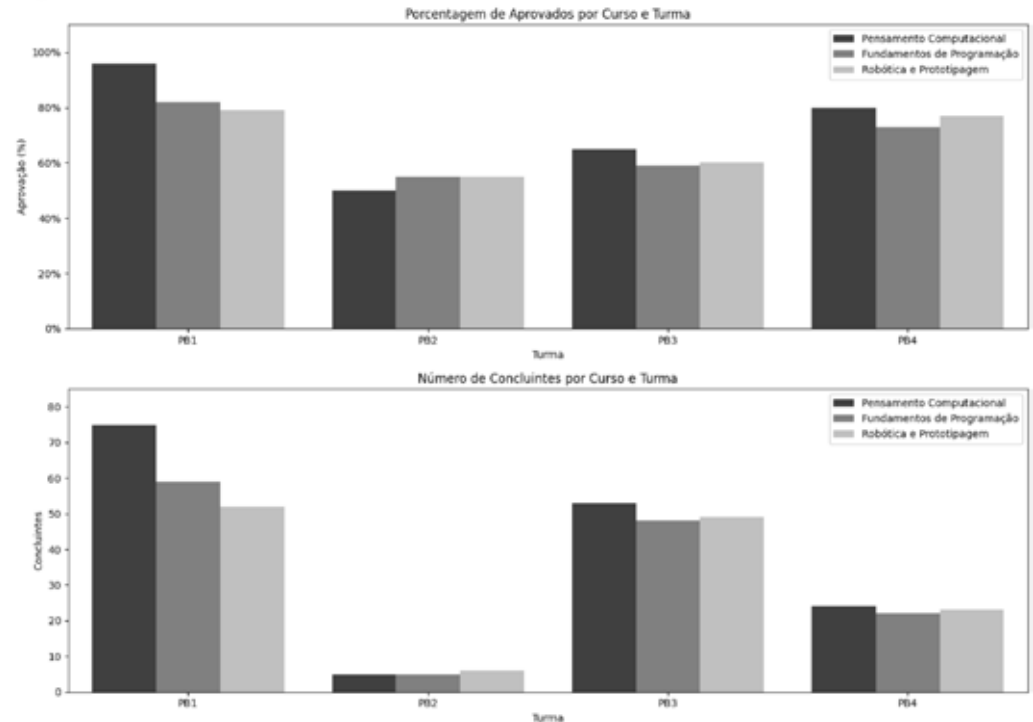
- No curso de Pensamento Computacional (PC), utiliza-se a programação em ícones, com a criação de histórias interativas por meio da ferramenta ScratchJr;
- No curso de Fundamentos de Programação (FP), adota-se a programação em blocos, com o desenvolvimento de jogos simples utilizando a plataforma Scratch;
- Por fim, no curso de Robótica e Prototipagem (RP), a sequência se encerra com o uso da programação híbrida, empregando a plataforma Make-Code em conjunto com o microcontrolador Micro:bit, possibilitando o controle de dispositivos físicos como LEDs e servomotores.

Na Figura 1 apresentam-se os gráficos com os resultados das análises quantitativas realizadas nos cursos de Pensamento Computacional, Fundamentos de Programação e Robótica e Prototipagem. As análises foram condu-

zidas, conforme mencionado anteriormente, nos municípios de Pato Branco (PB) e Londrina (LD). Foram consideradas duas dimensões principais:

- Taxa de aprovação por curso: Aprovados/(Aprovados+Reprovados) (desistentes excluídos);
- Concluintes (n): contagem por turma e curso (indicador de volume, não de desempenho).

Figura 1. Resultados das turmas analisadas na cidade de Pato Branco



Em Pato Branco, a leitura conjunta de % de aprovação e concluintes (n) indica declínio inicial na passagem da T1 (pré) para T2–T3 (pós), seguido de estabilização na T4. Essa trajetória aponta associação entre a adoção da sequência e maior coesão interna da trilha (PC⇔FD⇔RP), ao mesmo tempo em que evidencia o papel de fatores operacionais concomitantes que modulam as taxas absolutas no curto prazo.

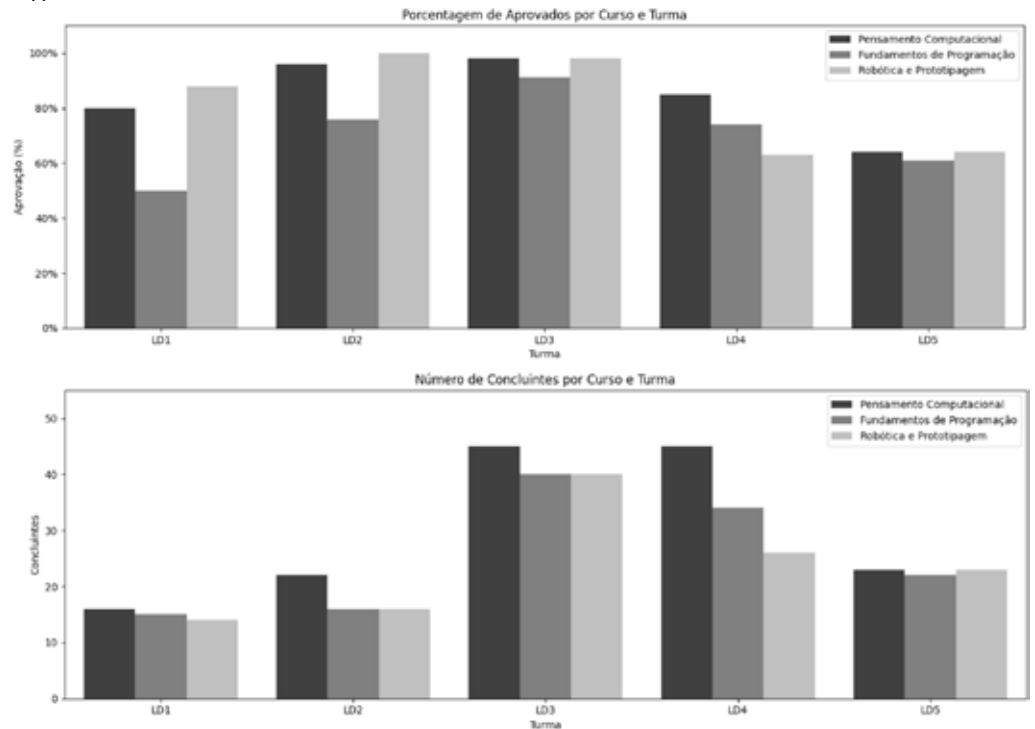
Em termos pedagógicos, a progressão não se traduz imediatamente em ganho percentual agregado, mas reduz a ruptura entre componentes à medida que a implementação se estabiliza e os estudantes transitam gradualmente para a programação textual.

Na T4 observa-se equilíbrio entre PC, FD e RP, sugerindo menor descontinuidade entre etapas e maior fluidez na transição. Tais achados estão em consonância com os fundamentos da teoria histórico-cultural de Vygotsky (1991), que destaca o papel das ferramentas simbólicas na mediação da aprendizagem, e com os princípios do construcionismo de Papert (1980), que valorizam a criação significativa como motor da aprendizagem ativa.

Além disso, os efeitos observados são corroborados por estudos revisados na literatura, como os de Lima Filho (2020) e Brito, Luz e Gonzalez (2018), que apontam que abordagens lúdicas e visuais no ensino da programação

promovem não apenas melhor desempenho acadêmico, mas também maior engajamento e motivação dos estudantes. A experiência em Pato Branco confirma essas evidências, reforçando o potencial pedagógico de sequências didáticas alinhadas ao desenvolvimento cognitivo dos aprendizes. Continuamos nossa análise, apresentando na Figura 2 as informações para a cidade de Londrina.

Figura 2. Resultados das turmas analisadas na cidade de Londrina



A análise das figuras referentes às turmas de Londrina evidencia que a adoção da sequência didática progressiva resultou em um nivelamento mais consistente da curva de aprendizagem, especialmente nas transições entre os cursos de Pensamento Computacional, Fundamentos de Programação e Robótica e Prototipagem. Essa melhoria torna-se perceptível a partir da Turma LD4, quando a proposta metodológica passou a ser implementada de forma estruturada.

A análise intra-turma, que observa a variação das taxas de aprovação entre os componentes dentro de uma mesma turma, evidencia tendências importantes. Em Pato Branco, a diferença entre PC e FP diminuiu progressivamente de -14,4 pontos percentuais na turma T1 para +0,3 p.p. na turma T4, enquanto a diferença entre FP e RP passou de -3,2 p.p. para +0,2 p.p. Esse comportamento demonstra uma redução gradual das rupturas entre disciplinas, até alcançar um equilíbrio entre as etapas da trilha formativa. Em Londrina, a tendência é parcialmente distinta: as turmas T1 a T3 (pré) já apresentavam pequenas quedas de PC para FP (-9 a -11 p.p.) e forte recuperação em RP (+9 a +18 p.p.), enquanto nas turmas T4 e T5 (pós) a transição entre PC e FP manteve-se negativa (-5 a -10 p.p.) e a etapa de RP apresentou queda adicional (-2 a -7 p.p.). Esses resultados sugerem que, em contextos mais maduros e estabilizados, a progressão tende a atenuar a queda entre

componentes, enquanto em períodos de implantação e reorganização estrutural os efeitos são temporariamente mitigados.

Tais dados sugerem que a progressão planejada de linguagens de programação, de visuais à textuais com experimentação prática, respeita os estágios de desenvolvimento cognitivo e favorece, de maneira significativa, a compreensão dos conteúdos, a permanência dos estudantes e a fluidez no percurso formativo.

Essas variáveis estruturais explicam as discrepâncias entre o comportamento quantitativo das taxas e a evolução qualitativa do processo de aprendizagem. Os resultados intra-turma revelam que, nas turmas mais recentes, a progressão favoreceu a coesão entre os módulos, diminuindo a perda de desempenho na passagem de uma etapa a outra. Em Pato Branco, o modelo atingiu estabilidade na turma T4, com equivalência de desempenho entre PC, FP e RP; em Londrina, a heterogeneidade de formato e a implementação parcial do modelo explicam a persistência das diferenças.

Os resultados obtidos nos municípios de Pato Branco e Londrina encontram respaldo em evidências científicas previamente publicadas. Lima Filho (2020), em sua pesquisa intitulada “Ensino e aprendizagem de lógica de programação com linguagem visual em blocos no 5º ano do ensino fundamental”, conduziu um estudo envolvendo estudantes que foram introduzidos à lógica de programação utilizando uma linguagem visual baseada em blocos. A pesquisa teve duração de oito semanas, com sessões semanais de 90 minutos.

De forma complementar, a revisão sistemática realizada por Noone e Mooney (2018), no artigo “Visual and Textual Programming Languages: A Systematic Review of the Literature” analisou diversos estudos comparando a eficácia de linguagens visuais e textuais no ensino de programação para iniciantes. Os autores constataram que estudantes que iniciam sua aprendizagem com linguagens visuais, como o Scratch, desenvolvem uma compreensão conceitual mais consistente, enquanto aqueles que começam diretamente com linguagens textuais apresentam maior dificuldade de abstração, o que frequentemente se traduz em altas taxas de evasão em cursos de introdução à computação.

Por fim, destaca-se o estudo “O Uso de Ferramentas Lúdicas para o Aprendizado de Programação com Jovens do Ensino Fundamental II” (Brito; Luz; Gonzalez, 2018), conduzido com duas turmas do Ensino Fundamental II em uma escola pública de Alagoinhas, Bahia. Uma turma foi submetida ao ensino tradicional de lógica de programação, enquanto a outra utilizou o Scratch, com sessões semanais de 60 minutos.

Os estudos analisados fornecem evidências, tanto qualitativas quanto quantitativas, que sustentam a eficácia de uma abordagem progressiva no ensino de programação. Iniciar o processo de aprendizagem com ferramentas lúdicas e visuais, baseadas em linguagens icônicas e em blocos, favorece a compreensão conceitual inicial, ao mesmo tempo em que promove maior engajamento e motivação dos estudantes.

À medida que os aprendizes se familiarizam com os princípios fundamentais da lógica computacional, a transição para linguagens textuais ocorre de forma mais fluida e menos intimidante, permitindo a consolidação de competências cognitivas e promovendo uma trajetória de aprendizagem mais efetiva, significativa e sustentável.

Sob uma perspectiva científica, a análise dos dados corrobora a hipótese central deste estudo: a sequência didática progressiva, gera ganhos de coesão do percurso e atenua rupturas entre componentes sob condições operacionais adequadas; efeitos absolutos nas taxas tendem a emergir com a estabilização da implementação.

5. Considerações Finais

A análise desenvolvida neste artigo demonstrou que a implementação progressiva de ferramentas lúdicas no ensino-aprendizagem de programação constitui não apenas uma alternativa metodológica viável, mas uma necessidade pedagógica contemporânea, fundamentada em teorias sólidas da aprendizagem e respaldada por evidências empíricas consistentes.

A fundamentação teórica ancorada na Pedagogia das Competências, bem como nas contribuições de Piaget e Vygotsky, reforçou a importância de respeitar os processos naturais do estudante, utilizando ferramentas que favoreçam a construção gradual do pensamento lógico-computacional. A sequência didática, mostrou-se mais adequada ao desenvolvimento das competências requeridas para a formação crítica, criativa e autônoma dos aprendizes.

Os dados empíricos analisados, principalmente os oriundos do Projeto Letramento Digital da FACTI, reforçaram esta tese. A experiência prática nos municípios de Pato Branco e Londrina evidenciou que a adoção estruturada da progressão pedagógica resultou em melhorias significativas na curva de aprendizagem, redução das taxas de desistência e maior estabilidade no desempenho entre as etapas formativas. A comparação entre turmas submetidas à progressão didática e aquelas inseridas na abordagem tradicional confirmou a eficácia da proposta e validou a hipótese central deste estudo.

Adicionalmente, os estudos de caso revisados na literatura internacional e nacional corroboraram que o uso de linguagens de programação visuais e progressivas favorece o engajamento, a retenção do conhecimento e o desenvolvimento de habilidades lógicas, superando as limitações das práticas que iniciam diretamente com linguagens textuais estruturadas.

Em síntese, os dados permitem responder às três questões de pesquisa de forma articulada. No que se refere à RQ1, os resultados confirmam a atenuação da queda entre componentes em contextos estabilizados (notadamente em Pato Branco), reforçando o papel da progressão na continuidade cognitiva e conceitual entre as disciplinas. Houve atenuação da quebra PC_FP e FP_RP em Pato Branco (pré: -14,4/-3,2 p.p.; pós: -4,2/+2,2 p.p.). Em Londrina, o efeito foi parcial (leve melhora PC_FP; perda em RP).

Por fim, a RQ2 é amplamente corroborada: o efeito da progressão é condicional ao contexto de implementação, sendo potencializado pela existência de tempo de prática protegido, acompanhamento síncrono, infraestrutura adequada e rubricas avaliativas progressivas, e limitado por formatos predominantemente assíncronos, sobrecarga de turmas e descontinuidade docente.

Diante desses achados, recomenda-se que projetos educacionais de formação em tecnologia e programação, seja na Educação Básica ou na formação técnico-profissional, adotem a sequência didática progressiva aqui analisada. Além disso, ressalta-se a urgência de políticas públicas que viabilizem a formação continuada de professores nesse novo paradigma pedagógico, bem como a ampliação do acesso equitativo a tecnologias educativas compatíveis com a realidade das escolas públicas brasileiras.

5.1 Perspectivas futuras e limitações

Apesar dos resultados positivos apresentados, é importante reconhecer que a interpretação dos dados carece de uma análise crítica mais aprofundada sobre fatores externos que podem ter influenciado os resultados além da progressão didática. O estudo não contempla a aplicação de testes estatísticos robustos, como análise de variância, cálculo de margens de erro ou intervalos de confiança, o que reduz a força quantitativa das evidências apresentadas. Recomenda-se que futuras pesquisas incorporem critérios estatísticos mais rigorosos, tabelas descritivas completas e delineamentos experimentais que possibilitem comparações mais precisas entre diferentes grupos educacionais.

Também é fundamental que investigações subsequentes ampliem o escopo da análise, incluindo estudos longitudinais e interseccionais que considerem os efeitos da progressão didática sobre outras competências transversais, como a criatividade, a autonomia no uso das tecnologias digitais e a capacidade de resolução de problemas complexos.

Essas perspectivas poderão contribuir de forma significativa para o fortalecimento da base científica que sustenta a inovação educacional no campo da computação e do pensamento computacional, além de proporcionar evidências mais robustas para a formulação de políticas públicas e práticas pedagógicas eficazes.

6. Agradecimento

Este projeto foi apoiado pelo programa PPI Softex, Convênio/Acordo de Parceria nº 0200-03/2022/SOFTEX/FACTI/LETRAMENTODIGITAL, financiado pelo Ministério da Ciência, Tecnologia e Inovação com recursos da Lei nº 8.248, de 23 de outubro de 1991. Os autores agradecem respeitosamente a todas as pessoas e organizações que, direta ou indiretamente, contribuíram para a execução do projeto Letramento Digital, destacando, em particular,

todo o corpo dirigente e de servidores do Ministério da Ciência, Tecnologia e Inovação.

7. Referências

- AUSUBEL, D. P. Aquisição e retenção de conhecimentos: uma perspectiva cognitiva. Lisboa: Plátano, 2003.
- BAU, D.; GRAY, J.; KELLEHER, C.; SHELDON, J.; TURBAK, F. Learnable programming: blocks and beyond. *Communications of the ACM*, New York, v. 60, n. 6, p. 72–80, 2017. DOI: <https://doi.org/10.1145/3015455>.
- BRITO, E. S.; LUZ, K. P.; GONZALEZ, L. S. O uso de ferramentas lúdicas para o aprendizado de programação com jovens do Ensino Fundamental II. *Revista Científica Multidisciplinar Núcleo do Conhecimento*, v. 3, n. 5, p. 14–25, 2018. Disponível em: https://www.cairu.br/revista/arquivos/artigos/20182/art_01.pdf. Acesso em: 13 abr. 2025.
- [CASAGRANDA, M.; FREITAS, P. F. O uso de ferramentas de programação visual no ensino de programação no ensino médio. SciELO Preprints, 2023. Preprint. Disponível em: <https://doi.org/10.1590/SciELOPreprints.11081>. Acesso em: 6 nov. 2025.](#)
- [COSTA, G. R.; KOLOGESKI, A. L. Revisão sistemática sobre o ensino de programação. Publica-IFRS: Boletim de Pesquisa e Inovação, Bento Gonçalves, v. 2, n. 1, p. 1–9, 2024. Disponível em: <https://doi.org/10.35819/publica-ifrs.v2.n1.a7508>. Acesso em: 6 nov. 2025.](#)
- [DAHALAN, F.; ALIAS, N.; SHAHAROM, M. S. N. Gamification and Game Based Learning for Vocational Education and Training: A Systematic Literature Review. Education and Information Technologies, Dordrecht, 2023/2024. Disponível em: <https://doi.org/10.1007/s10639-022-11548-w>. Acesso em: 6 nov. 2025.](#)
- [KELLEHER, C.; PAUSCH, R. Lowering the barriers to programming: a taxonomy of programming environments and languages for novice programmers. ACM Computing Surveys, New York, v. 37, n. 2, p. 83–137, 2005. DOI: <https://doi.org/10.1145/1089733.1089734>.](#)
- [LIMA, J. R.; MENEZES, C. S. As dificuldades enfrentadas pelos estudantes na aprendizagem de programação de computadores: uma revisão sistemática da literatura. RENOTE – Revista Novas Tecnologias na Educação, Porto Alegre, v. 22, n. 1, p. 130–140, 2024. Disponível em: <https://doi.org/10.22456/1679-1916.141538>. Acesso em: 6 nov. 2025.](#)
- [LIMA FILHO, M. P. Ensino e aprendizagem de lógica de programação com linguagem visual em blocos no 5º ano do ensino fundamental. 2020. 65 f. Trabalho de Conclusão de Curso \(Bacharelado em Sistemas de Informação\) – Unidade Acadêmica de Educação a Distância e Tecnologia, Universidade Federal Rural de Pernambuco, Recife, 2020. Disponível em: <https://repository.ufrpe.br/handle/123456789/2210>. Acesso em: 13 abr. 2025.](#)

- [LUXTON-REILLY, A. Learning to Program is Easy. In: ANNUAL CONFERENCE ON INNOVATION AND TECHNOLOGY IN COMPUTER SCIENCE EDUCATION \(ITicSE\), 21., 2016, Arequipa. Proceedings... New York: ACM, 2016. p. 284–289. DOI: <https://doi.org/10.1145/2899415.2899432>.](#)
- [MESSER, M.; BROWN, N. C. C.; KÖLLING, M.; SHI, M. Automated grading and feedback tools for programming education: a systematic review. arXiv, 2023. Disponível em: <https://arxiv.org/abs/2306.11722>. Acesso em: 27 mar. 2025.](#)
- [MICRO:BIT EDUCATIONAL FOUNDATION. Impact report 2019. United Kingdom: Micro:bit Educational Foundation, 2019. Disponível em: <https://microbit.org/research/>. Acesso em: 27 mar. 2025.](#)
- [NOONE, M.; MOONEY, A. Visual and textual programming languages: a systematic review of the literature. Journal of Computers in Education, v. 5, n. 2, p. 149–174, 2018. DOI: <https://doi.org/10.1007/s40692-018-0101-5>.](#)
- [OLIVEIRA, M. D. Programação com crianças: Scratch e o raciocínio lógico. 2013. Dissertação \(Mestrado em Educação\) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013. Disponível em: <https://lume.ufrgs.br/handle/10183/74830>. Acesso em: 27 mar. 2025.](#)
- [PAN, Y.; HARUN, J. Conquering Coding Fears: A Systematic Review of Programming Anxiety in Higher Education. Journal of Information Technology Education: Research, 2025. Disponível em: <https://doi.org/10.28945/5543>. Acesso em: 6 nov. 2025.](#)
- [PAPERT, S. Mindstorms: children, computers, and powerful ideas. New York: Basic Books, 1980.](#)
- [PERRENOUD, P. Construir as competências desde a escola. Porto Alegre: Artmed, 1999.](#)
- [PERRENOUD, P. Construir as competências desde a escola. Porto Alegre: Artmed, 2000.](#)
- [PIAGET, J. A equilibração das estruturas cognitivas. Rio de Janeiro: Zahar, 1975.](#)
- [PIAGET, J. A formação do símbolo na criança: imitação, jogo e sonho, imagem e representação. Rio de Janeiro: Zahar, 1976.](#)
- [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 all. Communications of the ACM, New York, v. 52, n. 11, p. 60–67, 2009. DOI: <https://doi.org/10.1145/1592761.1592779>](#)
- [ROBINS, A.; ROUNTREE, J.; ROUNTREE, N. Learning and teaching programming: a review and discussion. Computer Science Education, v. 13, n. 2, p. 137–172, 2003. DOI: <https://doi.org/10.1076/csed.13.2.137.14200>.](#)

UNAL, A.; TOPU, F. B. Effects of teaching a computer programming language via hybrid interface on anxiety, cognitive load level and achievement of high school students. Education and Information Technologies, v. 26, p. 5291–5309, 2021. Disponível em: <https://doi.org/10.1007/s10639-021-10536-w>. Acesso em: 6 nov. 2025.

VIDENOVIK, M.; VOLD, T.; KIØNIG, L.; MADEVSKA BOGDANOVA, A.; TRAJKOVIK, V. Game-based learning in computer science education: a scoping literature review. International Journal of STEM Education, 10:54, 2023. Disponível em: <https://doi.org/10.1186/s40594-023-00447-2>. Acesso em: 6 nov. 2025.

VYGOTSKY, L. S. A formação social da mente: o desenvolvimento dos processos psicológicos superiores. São Paulo: Martins Fontes, 1991.

WING, Jeannette M. Computational thinking. Communications of the ACM, v. 49, n. 3, p. 33–35, 2006. DOI: <https://doi.org/10.1145/1118178.1118215>.

ZABALA, A.; ARNAU, L. Como aprender e ensinar competências. Porto Alegre: Artmed, 2010.