

Sistemas de Aprendizagem Adaptativa aplicados ao Ensino de Lógica de Programação: uma revisão sistemática da literatura

Lucineide Nunes Pimenta, UNIFESP, lucineide.pimenta@unifesp.br,
<https://orcid.org/0000-0001-6850-6946>

André Fernando de Almeida, UNIFESP, andre.almeida29@unifesp.br,
<https://orcid.org/0000-0002-8211-599X>

Camila Bertini Martins, UNIFESP, cb.martins@unifesp.br,
<https://orcid.org/0000-0002-8252-8815>

Flavia Cristina Martins Queiroz Mariano, ITA, flaviaqz@ita.br,
<https://orcid.org/0000-0002-0963-702X>

Mariana Aparecida de Oliveira Rosa, IFSP, mariana.rosa@ifsp.edu.br,
<https://orcid.org/0009-0001-8241-536X>

Tiago de Oliveira, UNIFESP, tiago.oliveira@unifesp.br,
<https://orcid.org/0000-0002-3676-5967>

Resumo: Este artigo apresenta uma Revisão Sistemática da Literatura (RSL) sobre a aplicação de Sistemas de Aprendizagem Adaptativa (SAA) no ensino de Lógica de Programação (LP). Com base no protocolo de Kitchenham, foram analisados 15 artigos publicados após triagem progressiva de 329 estudos iniciais. Os resultados indicam predomínio de soluções voltadas ao ensino superior, com equilíbrio tecnológico entre algoritmos de Aprendizado de Máquina e mecanismos baseados em regras heurísticas. Os principais impactos observados consistem na melhora do desempenho acadêmico, maior engajamento e motivação estudantil. Conclui-se que o avanço da área requer estudos longitudinais, maior inserção do corpo docente no desenvolvimento das ferramentas e o alinhamento com abordagens cognitivas cientificamente validadas.

Palavras-chave: sistemas de aprendizagem adaptativa; lógica de programação; revisão sistemática da literatura; tecnologia educacional.

Adaptive Learning Systems applied to Programming Logic Teaching: a systematic literature review

Abstract: This paper presents a Systematic Literature Review (SLR) on the application of Adaptive Learning Systems (ALS) in the teaching of Programming Logic (PL). Based on Kitchenham's protocol, 15 articles published were analyzed after a progressive screening of 329 initial studies. The results indicate a predominance of solutions aimed at higher education, with a technological balance between Machine Learning algorithms and mechanisms based on heuristic rules. The main impacts observed consist of improved academic performance, greater engagement, and student motivation. It is concluded that advancing the field requires longitudinal studies, greater involvement of the faculty in tool development, and alignment with scientifically validated cognitive approaches.

Keywords: adaptive learning systems; programming logic; systematic literature review; educational technology.

1. Introdução

A disciplina de Lógica de Programação (LP) configura-se como um dos principais pilares na formação de profissionais de ciência da computação e áreas correlatas. Todavia, a literatura especializada pontua que seu aprendizado é marcado por altos índices de evasão, reprovação e desmotivação estudantil (HOED, 2016; CORRÊA et al., 2019). As barreiras envolvem desde a transição do pensamento puramente abstrato para a formalização algorítmica até a rigidez metodológica tradicional que desconsidera as heterogeneidades cognitivas das turmas.



Nesse cenário, os Sistemas de Aprendizagem Adaptativa (SAA) ganham destaque ao propor a modelagem do perfil do estudante para fornecer percursos formativos customizados (BARBOSA et al., 2024). De acordo com a literatura, a Lógica de Recomendação implementada nesses sistemas atua como a espinha dorsal da personalização, conectando a coleta de dados educacionais às práticas de adequação de conteúdo (LUCENA et al., 2021; BARBOSA et al., 2024).

Diante da pluralidade de soluções emergentes, faz-se imperativo mapear o estado da arte para identificar quais estratégias vêm sendo adotadas e quais lacunas persistem. Este artigo reporta os achados de uma Revisão Sistemática da Literatura (RSL) dedicada a responder como os SAA têm sido aplicados especificamente ao ambiente de ensino-aprendizagem de Lógica de Programação.

2. Metodologia da RSL

O estudo seguiu o protocolo estruturado por Kitchenham (2004), devendo-se estabelecer as questões de pesquisa (subseção 2.1), a estratégia e mecanismos de busca (subseção 2.2), os critérios de inclusão e exclusão (subseção 2.3) e o processo de seleção dos estudos (subseção 2.4).

2.1. Questões de Pesquisa

A formulação das questões de pesquisa direcionou todas as etapas subsequentes, da estratégia de busca à análise realizada. O estudo foi guiado pelas seguintes questões (Q1-Q4):

- Q1 - Como se configuram os sistemas de aprendizagem adaptativa voltados para o ensino de lógica de programação? Busca mapear as características e propósitos gerais das soluções desenvolvidas na literatura.
- Q2 - Quais fundamentações teóricas estão sendo citadas para a aprendizagem adaptativa no ensino de lógica de programação? Investiga os referenciais pedagógicos e computacionais que amparam os modelos de personalização.
- Q3 - De que forma funcionam os sistemas de aprendizagem adaptativa descritos nos estudos analisados? Mapeia os algoritmos de recomendação, técnicas de IA e mecanismos de modelagem adotados pelos sistemas.
- Q4 - Como os sistemas de aprendizagem adaptativa voltados ao ensino de lógica de programação têm sido avaliados? Examina as metodologias, instrumentos e públicos-alvo considerados nas validações empíricas.

O alinhamento dessas quatro questões assegura uma cobertura abrangente do estado da arte, permitindo compreender desde os conceitos teóricos até a implementação prática e os métodos de avaliação dos sistemas adaptativos. Dessa forma, cria-se uma base sólida para a categorização e síntese dos dados extraídos da literatura.

2.2. Mecanismos e *String* de Busca

A definição dos mecanismos de busca baseou-se na representatividade das bases de dados para a área de Educação em Computação e Tecnologias Educacionais. As bases consultadas foram: ACM Digital Library, Google Scholar, IEEE Digital Library, SBC OpenLib (SOL), Scielo, Science Direct e Springer Link.

A busca foi conduzida de forma iterativa através de três eixos principais (Educação em Computação; Sistemas Adaptativos/Recomendação; e Estilos de Aprendizagem/Traços Pessoais). A *string* consolidada foi aplicada em maio de 2025, sendo ela: ("*computer science education*" OR

"teaching of programming logic") AND ("adaptive educational system" OR "e-learning system" OR "recommender system") AND ("learning styles" OR "personal traits").

2.3. Critérios de Inclusão/Exclusão, Avaliação de Qualidade (AQA) e Mapeamento de Citações

Os critérios de seleção foram divididos em etapas eliminatórias para garantir a precisão no recorte da amostra. Foram incluídos apenas trabalhos que descrevessem de forma direta o desenvolvimento ou aplicação de SAA voltados à disciplina de LP ou tópicos correlatos, aplicados a estudantes do ensino básico, médio ou universitário. Excluíram-se trabalhos em que a lógica de programação não fosse o eixo central de aprendizado, além de estudos não revisados por pares, de revisão prévios, resumos informais ou estudos redundantes de mesma autoria.

Após a aplicação dos filtros iniciais, os artigos elegíveis foram submetidos a um processo de Avaliação de Qualidade dos Artigos (AQA). A AQA consistiu em um conjunto de critérios ponderados baseados nas diretrizes de Kitchenham, avaliando seis eixos principais: (1) clareza dos objetivos de pesquisa; (2) explicitação do nível de ensino e do público-alvo; (3) fundamentação algorítmica ou arquitetural do sistema adaptativo; (4) detalhamento dos objetos ou materiais de aprendizagem utilizados; (5) robustez da metodologia de validação (ex: presença de grupo de controle ou pré/pós-testes); e (6) discussão explícita das limitações do estudo.

Cada critério foi pontuado em uma escala de aderência (0 para não atende, 0,5 para atende parcialmente e 1,0 para atende integralmente). A nota final de qualidade foi convertida em uma escala percentual. Estabeleceu-se uma nota de corte de 50% de aproveitamento na AQA; qualquer artigo com rigor inferior a este limite foi excluído para preservar o valor empírico do corpus documental.

Por fim, o número de citações de cada artigo foi registrado, sendo mensurado de forma padronizada por meio do agregador do Google Scholar e capturado no mês de maio de 2025.

2.4. Processo de Seleção (Fluxo de Estudos)

O processo seguiu uma triagem progressiva em múltiplos níveis, ilustrada nas etapas sequenciais abaixo:

1. **Identificação:** Execução da *string* nas 7 bases de dados selecionadas, resultando na recuperação de 329 artigos (ACM Digital Library: 5, Google Scholar: 28, IEEE Digital Library: 133, SBC OpenLib (SOL): 66, Scielo: 2, Science Direct: 52 e Springer Link: 43).
2. **Remoção de Duplicatas:** Tratamento de metadados via StArt e Zotero, resultando na exclusão de 5 duplicatas (restando 324 títulos).
3. **Triagem por Título/Palavras-Chave:** Filtro preliminar que eliminou 290 estudos desalinhados, deixando 34 artigos elegíveis.
4. **Elegibilidade (Texto Completo):** Leitura integral dos 34 artigos remanescentes, com aplicação dos critérios detalhados, culminando na exclusão de 21 artigos.
5. **Inclusão Adicional (Snowballing):** Aplicação do método bola de neve (*snowballing*) a partir das referências dos artigos mapeados, integrando novos estudos relevantes até consolidar o corpus documental definitivo de 15 trabalhos para a análise final.

Ao término dessa triagem, consolidou-se um corpus final de 15 estudos primários diretamente aderentes aos objetivos da pesquisa. Essa seleção delimitou a base documental que sustenta as análises qualitativas e quantitativas apresentadas nas seções subsequentes.



3. Resultados e Discussão

3.1. Panorama Geral (Q1)

A análise cronológica e o mapeamento de relevância científica revelam um campo dinâmico, marcado por publicações de alto impacto em paralelo a contribuições recentes de alto rigor metodológico. A Tabela 1 consolida os quinze artigos que compõem o escopo analítico final desta RSL, correlacionando o ano de publicação, o volume de interesse acadêmico por período, o volume absoluto de citações (apurado via Google Scholar em 2025) e o desempenho percentual alcançado na Avaliação de Qualidade (AQA).

A leitura dos dados contidos na Tabela 1 indica que o ano de 2018 representou o pico produtivo em volume absoluto de publicações, acumulando 4 artigos selecionados. Sob a perspectiva do impacto mapeado por meio da análise bibliométrica de citações do Google Scholar, o estudo de Sosnovsky e Brusilovsky (2015) destaca-se expressivamente como o maior marco de influência, acumulando 82 citações registradas.

Tabela 1 - Panorama cronológico, impacto acadêmico e avaliação de qualidade (AQA)

Ano	Qtd. Art.	Artigos Representativos do Ano (Citação)	Número de Citações Registrado (Scholar/2025)	Critério de Qualidade (AQA)
2013	1	Chrysafiadi e Virvou (2013)	56	86%
2015	2	Sosnovsky e Brusilovsky (2015)	82	86%
		Anitha e Deisy (2015)	20	86%
2016	1	Dogan e Dikbiyik (2016)	23	89%
2017	2	Yi et al. (2017)	3	93%
		Evale (2017)	24	89%
2018	4	Gavrilović et al. (2018)	34	71%
		Saito e Watanobe (2018)	34	57%
		Nabizadeh et al. (2018)	27	71%
		Santos et al. (2018)	4	100%
2019	2	Adnan et al. (2019)	16	82%
		Sogy et al. (2019)	2	57%
2020	1	Macena et al. (2020)	13	100%
2021	1	Neto et al. (2021)	4	100%
2024	1	Kumar e Thandeeswaran (2024)	3	100%

Ao cruzar os dados de impacto de citações com os índices da Avaliação de Qualidade (AQA), observa-se um amadurecimento metodológico na área. Os trabalhos publicados nos anos mais recentes da janela temporal investigada (2020, 2021 e 2024) alcançaram de forma unânime a pontuação máxima de 100% no critério de qualidade. Esse fenômeno demonstra que, embora os estudos novos — como o de Kumar e Thandeeswaran (2024) — ainda acumulem poucas citações devido ao curto tempo de exposição na área acadêmica, eles foram projetados e descritos sob desenhos experimentais mais rigorosos, explicitando com clareza a coleta de *logs*, a arquitetura dos algoritmos e abordagens multifacetadas de testes empíricos com grupos de controle.

Em contrapartida, alguns estudos de períodos intermediários, como Saito e Watanobe (2018), obtiveram as menores pontuações da amostra na AQA (57%), embora registrem um número expressivo de citações (31). A análise qualitativa revela que essa discrepância ocorre porque tais estudos priorizaram abordagens mais matemáticas ou algoritmos de IA de caixa-preta, sem amparo específico em teorias educacionais e omitindo discussões essenciais sobre limitações de implementação em infraestruturas reais e o papel do docente na mediação tecnológica.

Por sua vez, em relação ao público-alvo, constatou-se que a grande maioria dos SAA é desenvolvida para o ensino superior. Iniciativas voltadas à educação básica ocorrem de forma bem

reduzida, amparadas por dinâmicas lúdicas em jogos escolares, como observado em Macena et al. (2020).

Na análise dos 15 artigos estudados, 12 inclui estudos voltados ao ensino superior: Neto et al. (2021), Santos et al. (2018), Kumar e Thandeeswaran (2024) e Yi et al. (2017), assim como os estudos de Adnan et al. (2019), Anitha e Deisy (2015), Chrysafiadi e Virvou (2013), Dogan e Dikbiyik (2016), Evale (2017), Gavrilović et al. (2018), Nabizadeh et al. (2018) e Sosnovsky e Brusilovsky (2015) desenvolveram SAA para estudantes de graduação na área de Ciência da Computação. Apenas o trabalho de Macena et al. (2020) incluiu como público-alvo do seu SAA crianças com idade entre 8 e 11 anos, configurando-se como o único a contemplar esse público. Em dois casos, não há clareza sobre o nível de ensino. Sogy et al. (2019) não especificam se os participantes eram de nível médio, superior ou técnico. Já Saito e Watanobe (2018) utilizam a denominação genérica “*students*” em um ambiente de *online judge*, sem explicitar o contexto educacional.

Ainda sobre o público-alvo, prevaleceu a concepção de sistemas centrados no estudante, privilegiando a personalização e o progresso individual. No entanto, nos trabalhos em que o público duplo (estudantes e professores) é contemplado, observa-se uma tendência de valorização do *design* participativo. Santos et al. (2018) e Neto et al. (2021), por exemplo, destacam o envolvimento de docentes na concepção das atividades, mesmo que a execução seja centrada nos estudantes. Macena et al. (2020) também inserem o corpo docente no processo de desenvolvimento, o que contribui para alinhar o conteúdo algorítmico às necessidades pedagógicas. Já Yi et al. (2017) permitem que instrutores configurem trilhas de aprendizagem, enquanto Kumar e Thandeeswaran (2024) desenvolvem um SAA aplicado a vídeos de programação, em que instrutores podem assumir o papel de mediadores de conteúdo adaptativo.

Em contrapartida, um conjunto significativo de estudos focaliza exclusivamente os estudantes como usuários finais. É o caso de Sosnovsky e Brusilovsky (2015), cujo sistema se restringe a avaliações de aprendizagem; Anitha e Deisy (2015) e Gavrilović et al. (2018), que não mencionam a figura docente; além de Saito e Watanobe (2018) e Sogy et al. (2019), que projetam ambientes voltados apenas à prática estudantil. Vale pontuar também o estudo de Chrysafiadi e Virvou (2013) que detalha o uso do sistema pelos estudantes, mas não explicita a participação docente. Situação semelhante ocorre em Evale (2017), onde relatórios são mencionados como possíveis ferramentas de acompanhamento para professores, sem, contudo, caracterizá-los como usuários ativos. Dogan e Dikbiyik (2016), embora concentrem a aplicação nos estudantes, reconhecem a função dos especialistas na configuração dos conteúdos. Já em Adnan et al. (2019) e Nabizadeh et al. (2018), a referência ao corpo docente é praticamente ausente, o que evidencia a predominância do foco discente.

Por fim, quanto aos conteúdos programáticos mapeados, a maior parte dos SAA foca em tópicos da lógica, tais como variáveis, constantes, condicionais, algoritmos e estruturas de dados básicas (*arrays/pointers*). Contudo, notou-se que os sistemas frequentemente vinculam o ensino da lógica à sintaxe de linguagens específicas (como Java ou C). Essa abordagem prioriza o mecanismo de recomendação adaptativa (como aprender) em detrimento de discussões estritamente curriculares (o que aprender).

Entre os trabalhos com descrição mais consistente, observa-se forte presença de conceitos básicos e recorrentes no ensino introdutório de LP. O jogo de Neto et al. (2021), por exemplo, utiliza blocos que representam variáveis, laços, condições e operações lógicas, elementos centrais para a formação do raciocínio computacional. De forma semelhante, Santos et al. (2018) estruturaram seu jogo a partir de tópicos clássicos como variáveis, procedimentos, estruturas de controle e laços. Yi et al. (2017) também contemplam variáveis, laços e condições em Java, reforçando o caráter lógico da disciplina. Já Kumar e Thandeeswaran (2024) utilizaram vídeos instrucionais para apresentar trechos codificados voltados à lógica de programação.



Nos estudos que ampliam a abordagem, há propostas mais abrangentes: Chrysafiadi e Virvou (2013), por exemplo, incluem estruturas de repetição, *arrays*, operadores e algoritmos; Macena et al. (2020) exploram conceitos de algoritmos e estruturas de dados iniciais (listas, pilhas, filas) em jogos educativos; e Dogan e Dikbiyik (2016) estruturam o sistema OPCOMITS com base em mapas conceituais, cobrindo toda a disciplina de LP. Além disso, estudos como os de Adnan et al. (2019) e Evale (2017) inserem exercícios práticos de programação em contextos reais, enquanto Gavrilović et al. (2018) têm como alvo direto a programação em Java, utilizada como meio para reforçar a lógica.

Por outro lado, alguns trabalhos tratam a LP de forma apenas tangencial. Em Anitha e Deisy (2015), o conteúdo é restrito a *arrays* e *pointers* em C, sem ligação explícita com fundamentos lógicos. Nabizadeh et al. (2018) utilizam dados de cursos de programação, mas não especificam conteúdos clássicos como estruturas condicionais ou *loops*. Em Saito e Watanobe (2018), a ênfase recai sobre trilhas de aprendizado em ambientes de *online judges*, sem detalhamento dos tópicos de LP. Sogy et al. (2019) apresentam Python em tarefas genéricas de programação, sem conexão direta com variáveis ou estruturas de controle. E, finalmente, Sosnovsky e Brusilovsky (2015) abordam *quizzes* em Java e SQL, mas não tratam da lógica de forma direta.

3.2. Aportes e Fundamentações Teóricas (Q2)

Identificou-se uma rica heterogeneidade nos referenciais teóricos que dão suporte aos algoritmos adaptativos. A distribuição teórica dos pilares mapeados, juntamente com o foco operacional principal e as respectivas referências empíricas de cada estudo da amostra, encontra-se consolidada na Tabela 2.

Tabela 2 - Distribuição das teorias pedagógicas adotadas e estudos correspondentes

Referencial Teórico / Pedagógico	Freq.	Foco de Aplicação no Sistema Adaptativo	Referências dos Estudos Encontrados
Modelo de Felder-Silverman	3	Classificação em dimensões cognitivas (Ex: Ativo/Reflexivo, Sequencial/Global) para envio de objetos de aprendizagem.	Anitha e Deisy (2015); Evale (2017); Yi et al. (2017).
Modelo VARK	2	Divisão de preferência de recepção de conteúdo por canais (Visual, Auditivo, Leitura, Cinestésico).	Sogy et al. (2019); Kumar e Thandeeswaran (2024).
Sistemas de Tutoria Inteligente (ITS)	3	Modelagem cognitiva contínua do estudante, diagnóstico semântico e sintático automatizado de erros e uso de mapas conceituais.	Chrysafiadi e Virvou (2013); Sosnovsky e Brusilovsky (2015); Dogan e Dikbiyik (2016).
Aprendizagem Móvel (M-Learning)	1	Análise do contexto do estudante em tempo real (como localização, horário e conhecimentos prévios) para a recomendação.	Adnan et al. (2019).
Gamificação e Level Design Procedural	3	Adaptação em tempo real de mecânicas lúdicas, geração de caminhos em labirintos e roteiros instrucionais por nível.	Santos et al. (2018); Macena et al. (2020); Neto et al. (2021).
Abordagem Exclusivamente Computacional	3	Otimização e convergência de modelos matemáticos de IA, redes neurais e outras técnicas computacionais, sem amparo teórico educacional explícito.	Saito e Watanobe (2018); Gavrilović et al. (2018); Nabizadeh et al. (2018).

Um dos principais pilares de sustentação pedagógica para a personalização reside nos Modelos de Estilos de Aprendizagem (Felder-Silverman e VARK), presentes em cinco estudos da amostra, demonstrando forte relevância em exercerem uma ponte conceitual entre as características individuais psicométricas do aluno e a parametrização dos repositórios de objetos

de aprendizagem. Outros aportes recorrentes incluem os Sistemas de Tutoria Inteligente (ITS) e a Modelagem do Estudante baseada em dependências conceituais. Abordagens de cunho lúdico e construtivista também ganham relevância, amparadas por teorias de Gamificação, Retórica Procedural e *Design* Instrucional (*Level Design*) para modular desafios digitais em tempo real. Sendo assim, nota-se que o avanço recente das técnicas de jogos e gamificação — como visto em Santos et al. (2018) e Macena et al. (2020) — gerou uma nova vertente de SAA, cuja teoria pedagógica subjacente se apoia no construtivismo e no fluxo (*flow*) de desafios balanceados e adaptados contextualmente. Em contrapartida, uma parcela minoritária dos estudos, como o de Saito e Watanobe (2018), priorizou implementações exclusivamente matemáticas de Inteligência Artificial ou de outras técnicas computacionais, sem fundamentação pedagógica explícita.

3.3. Funcionamento e Mecanismos de Recomendação (Q3)

O funcionamento interno dos SAA revela a coexistência de arquiteturas clássicas (estruturadas em regras) e sistemas probabilísticos complexos. Para mapear o percurso do estudante a partir de um diagnóstico cognitivo, os sistemas costumam utilizar o fluxo lógico apresentado na Figura 1.

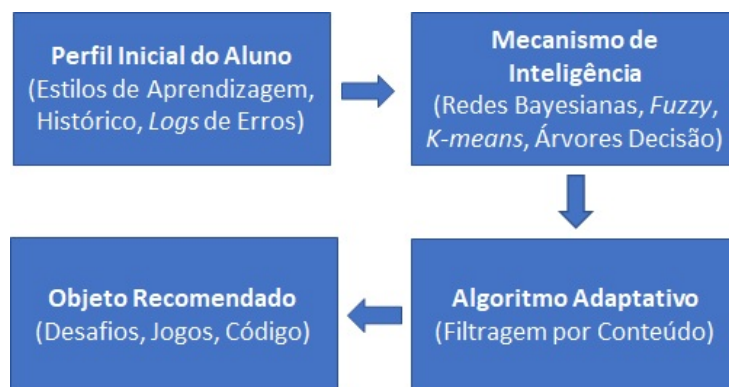


Figura 1 - Arquitetura padrão de funcionamento e recomendação dos SAA analisados

Quanto ao *Perfil Inicial* apresentado na Figura 1, a modelagem do estudante é uma etapa importante para o funcionamento de SAA: os sistemas capturam dados que variam de históricos de exercícios e desempenho em testes a traços comportamentais específicos e perfis cognitivos, utilizando essas informações para reordenar objetos de aprendizagem (*quizzes*, códigos, textos, vídeos) em tempo real. Nesse contexto, a literatura demonstra uma pluralidade de abordagens operacionais que se dividem principalmente a partir de duas perspectivas de modelagem do estudante:

1. **Modelagem Pedagógica e Cognitiva:** Os sistemas fundamentam e adaptam as trilhas pedagógicas, jogos ou materiais com base em teorias clássicas de *design* instrucional, estilos cognitivos estruturados de preferência dos estudantes ou mapas de dependência conceitual do domínio. Neste grupo, pode-se citar: Chrysafiadi e Virvou (2013); Anitha e Deisy (2015); Sosnovsky e Brusilovsky (2015); Dogan e Dikbiyik (2016); Santos et al. (2018); Macena et al. (2020) e Neto et al. (2021).
2. **Abordagens Baseadas em Dados (*Data-Driven*):** A modelagem e as recomendações acontecem de maneira responsiva, preditiva ou estatística a partir do consumo intensivo dos *logs* de desempenho e comportamento do discente no ambiente, muitas vezes operando sem a necessidade de um modelo teórico pedagógico explícito. Neste grupo, pode-se citar: Yi et al. (2017); Evale (2017); Gavrilović et al. (2018); Nabizadeh et al. (2018); Saito e Watanobe (2018); Sogy et al. (2019); Adnan et al. (2019) e Kumar e Thandeeswaran (2024).



Quanto ao *Mecanismo de Inteligência* e ao *Algoritmo Adaptativo* apresentado na Figura 1, os artigos identificados na literatura podem ser classificados de acordo com as seguintes categorias tecnológicas (Tabela 3):

1. **Modelos Probabilísticos e de Grafos:** Esta categoria reúne abordagens que lidam com a modelagem do domínio do conhecimento por meio de mapas conceituais ou árvores estruturadas, bem como o tratamento de incertezas nos perfis através de modelos numéricos e lógicos dinâmicos (como a *Lógica Fuzzy*). Operacionalmente, esses sistemas atualizam o nível de proficiência do discente à medida que ele resolve desafios associados a tópicos curriculares de programação (ex: variáveis, laços de repetição), permitindo avançar apenas quando o modelo calcula uma probabilidade de proficiência satisfatória.
2. **Classificadores e Aprendizado de Máquina:** Abrange os sistemas de alta complexidade computacional focados em técnicas *data-driven* (orientadas a dados). Eles operam capturando *logs* comportamentais detalhados, submetendo os dados históricos a algoritmos de agrupamento (*clustering*), árvores de decisão (como J48) ou séries temporais baseadas em redes neurais profundas (RNN). O objetivo operacional é classificar automaticamente o perfil cognitivo do aluno ou prever preditivamente qual o próximo exercício ideal para otimizar sua rota de aprendizagem.
3. **Mecânicas de Regras Adaptativas Próprias:** Compreende abordagens que não utilizam modelos complexos de *Machine Learning*, mas sim regras heurísticas estruturadas e mecanismos embutidos na lógica de programação da própria ferramenta. Destaca-se fortemente no subgrupo de jogos educacionais, plataformas de *quizzes* e ferramentas com lógicas de personalização, onde a modulação da dificuldade, a liberação de pistas e a progressão narrativa acontecem por meio de condicionais e regras de ativação fixadas previamente pelos criadores do software pedagógico.

Tabela 3 - Categorização tecnológica dos SAA identificados na RSL

Categoria Tecnológica	Técnicas Identificadas nos Artigos	Mecanismo Operacional e Aplicação Prática	Estudos Relacionados da RSL
Modelos Probabilísticos e de Grafos	(*) <i>Lógica Fuzzy</i> aplicada a ITS (*) Mapas Conceituais de Domínio (*) Modelagem de Incerteza de Caminhos	Estrutura dependências de pré-requisitos conceituais da linguagem e usa modelos estocásticos ou de inferência flexível para tratar o perfil dinâmico e medir a proficiência do aluno.	Chrysafiadi e Virvou (2013); Dogan e Dikbiyik (2016); Nabizadeh et al. (2018).
Classificadores e Aprendizado de Máquina	(*) Árvores de Decisão (J48, etc.) (*) Redes Neurais Recorrentes (RNN) (*) Algoritmos de Agrupamento (<i>Clustering</i>)	Consome os <i>logs</i> de uso do estudante em tempo real para classificar o comportamento ou prever preditivamente o próximo objeto de aprendizagem ou vídeo adequado.	Anitha e Deisy (2015); Yi et al. (2017); Evale (2017); Gavrilović et al. (2018); Saito e Watanobe (2018); Adnan et al. (2019); Kumar e Thandeeswaran (2024).
Mecanismos de Regras Adaptativas Próprias	(*) Regras Heurísticas por Tópicos (*) Retórica Procedural e Motores de Jogos (*) Lógicas de Personalização	Modula a entrega de conteúdos, <i>quizzes</i> , <i>feedbacks</i> e pistas lúdicas através de condicionais heurísticas embutidas na própria arquitetura de código ou narrativa do software.	Sosnovsky e Brusilovsky (2015); Santos et al. (2018); Sogy et al. (2019); Macena et al. (2020); Neto et al. (2021).

A categorização apresentada na Tabela 3 evidencia o equilíbrio da literatura: há um eixo voltado para Classificadores e Aprendizado de Máquina (7 estudos), enquanto soluções práticas



baseadas em Regras Próprias (5 estudos) e formulações matemáticas em Probabilidade e Grafos (3 estudos) complementam o panorama de SAA para o ensino de Lógica de Programação.

3.4. Métodos de Avaliação e Impactos Observados (Q4)

Para examinar a natureza das abordagens metodológicas, as métricas computacionais e os instrumentos de coleta de dados, a Tabela 4 detalha esse panorama, correlacionando os estudos da amostra às suas respectivas abordagens de avaliação e instrumentos de mensuração.

A análise da Tabela 4 revela uma clara ramificação no desenho metodológico das avaliações, dividindo-se entre validações de cunho estritamente tecnológico/algorítmico e experimentos focados no impacto pedagógico real em sala de aula. No primeiro grupo, destacam-se trabalhos como os de Saito e Watanobe (2018) e Nabizadeh et al. (2018), cujas abordagens priorizam a validação matemática de agrupamentos e a otimização de caminhos através de métricas de ciência de dados puras, como Silhouette Score, NDCG e MLL. Essas abordagens quantitativas puras convergem com as críticas em que o refinamento do algoritmo se sobrepõe ao debate curricular ou ao entendimento do processo cognitivo humano. Por outro lado, o corpus documental exibe um alinhamento com metodologias quase-experimentais e controladas para aferir o ganho de aprendizagem. Estudos como os de Dogan e Dikbiyik (2016) e Yi et al. (2017) utilizam o clássico modelo de avaliação diagnóstica estruturado em pré-testes e pós-testes aplicados a grupos de controle e experimental, recorrendo a testes de significância estatística para cancelar a evolução dos estudantes. Essa preocupação em comprovar empiricamente a eficácia da personalização pedagógica reflete o amadurecimento metodológico observado nos anos mais recentes da janela temporal investigada.

Ademais, a Tabela 4 evidencia o crescimento de abordagens multifacetadas, que buscam mensurar não apenas o ganho cognitivo isolado, mas a experiência do usuário (UX) e a aceitação tecnológica do SAA. O trabalho de Macena et al. (2020) ilustra essa tendência ao hibridizar a coleta automatizada de *logs* de interação com *frameworks* consagrados de avaliação de jogos (MEEGA+) e modelos de aceitação tecnológica (TAM).

Também vale pontuar que a maior parte dos estudos envolveu diretamente estudantes universitários, como nos trabalhos de Adnan et al. (2019), com 75 participantes no grupo experimental e 75 no grupo controle; no trabalho de Anitha e Deisy (2015) participaram 99 estudantes no estudo experimental com coleta de dados; no trabalho de Sogy et al. (2019), contou-se com 52 estudantes; e no trabalho de Neto et al. (2021) contou-se com 14 estudantes do ensino universitário. Embora em Macena et al. (2020) teve-se como foco o estímulo do raciocínio lógico de crianças, o sistema desenvolvido foi avaliado por 15 estudantes universitários.

Quanto aos impactos observados, os benefícios identificados nos estudos indicam melhora no desempenho em lógica de programação (ADNAN et al., 2019; YI et al., 2017), aumento de motivação (SANTOS et al., 2018), maior retenção de conteúdos (CHRYSAFIADI; VIRVOU, 2013) e engajamento em atividades lúdicas (MACENA et al., 2020; NETO et al., 2021). Além disso, a análise qualitativa identificou recorrência de termos como aprendizagem (8 menções), motivação (4), tempo (4) e personalização (4), indicando ênfase na adequação do ensino às necessidades individuais dos estudantes. Outros ganhos destacados foram eficiência, engajamento e desempenho (3 menções cada), além de evidências de maior compreensão conceitual (DOGAN; DIKBIYIK, 2016), e melhora em estruturas de controle e raciocínio lógico (SANTOS et al., 2018; NETO et al., 2021). Por outro lado, verificou-se que a grande maioria (12 dos 15 estudos) não relatou desvantagens ou limitações relevantes no uso de sistemas adaptativos.: Sosnovsky e Brusilovsky (2015) relataram desconfiança em recomendações consideradas excessivamente otimistas, Kumar e Thandeeswaran (2024) apontaram falhas na classificação de estilos de aprendizagem, e Evale (2017) mencionou frustração de alguns usuários em classificações equivocadas.

Tabela 4 - Detalhamento da avaliação dos sistemas e conteúdos recomendados

Autor (Ano)	Abordagem de Avaliação	Métricas e Instrumentos
Adnan et al. (2019)	Experimento comparativo em cenário real com grupos de estudantes usando dispositivos móveis (<i>M-learning</i>).	Análise computacional da acurácia do algoritmo de agrupamento (DBSCAN) + Questionário de percepção/satisfação contextualizada.
Anitha e Deisy (2015)	Validação prática da lógica de sequenciamento e agrupamento de Objetos de Aprendizagem (OAs) por similaridade.	Testes em situação real de sala de aula baseados em mapeamento por Ontologias e Estilos de Aprendizagem (Felder-Silverman).
Gavrilović et al. (2018)	Avaliação empírica do progresso do aprendizado por meio da execução de tarefas e exercícios práticos de programação.	<i>Logs</i> de interação e métricas algorítmicas de ajuste dinâmico de dificuldade com base na técnica de classificação KNN.
Chrysafiadi e Virvou (2013)	Medição contínua do progresso do aprendizado por meio de árvore de navegação em STL.	Teste de usabilidade em níveis incrementais de domínio e proficiência em programação, com aplicação do modelo estatístico MANOVA.
Dogan e Dikbiyik (2016)	Experimento controlado e comparativo baseado na aplicação de exames antes e após as intervenções educacionais.	Instrumentos de avaliação diagnóstica de conhecimento (Pré-teste e Pós-teste) aplicados a grupos controle e experimental.
Evale (2017)	Validação estatística de acurácia preditiva em relação ao rendimento e notas futuras projetadas para os discentes.	Testes padronizados de usabilidade de software combinados ao cálculo do coeficiente Kappa de concordância.
Macena et al. (2020)	Avaliação cobrindo simultaneamente a experiência de uso, satisfação do aluno e ganho real de aprendizado.	Modelos de avaliação de jogos MEEGA+ e modelo SAM (<i>Technology Acceptance</i>), aliados à extração automatizada de <i>logs</i> de interação.
Nabizadeh et al. (2018)	Validação algorítmica quantitativa focada na otimização e ajuste dos tempos e caminhos gerados pelo sistema.	Modelagem de análise estatística de erro focada em duas variáveis adaptativas: tempo de completude e pontuação (<i>score</i>).
Neto et al. (2021)	Avaliação incremental de software dividida em fases evolutivas de validação para garantir a integridade da recomendação.	Testes de aceitação com usuários em ambientes controlados organizados em ciclos sequenciais de versões Pré-Beta e Beta.
Saito e Watanobe (2018)	Validação matemática de agrupamento e ordenação algorítmica de exercícios no repositório de dados computacionais.	Métricas computacionais de avaliação de dados (<i>Silhouette Score</i> , <i>NDCG</i> e <i>MLL</i>).
Santos et al. (2018)	Aplicação em ambiente prático com foco no fluxo de execução de desafios de lógica pelos discentes.	Roteiros de dados observacionais coletados em tempo real combinados ao desempenho por fases e ao <i>feedback</i> direto dos estudantes.
Sogy et al. (2019)	Mapeamento e categorização dos padrões de comportamento e hábitos de estudo gerados durante a navegação.	Aplicação de <i>quizzes</i> rápidos de checagem integrados ao monitoramento contínuo de <i>logs</i> de acessos e eventos na plataforma.
Sosnovsky e Brusilovsky (2015)	Avaliação quantitativa sobre a efetividade pedagógica e a precisão da entrega das recomendações.	Métricas estatísticas de retenção de conhecimento, compreensão conceitual e progresso de desempenho.
Kumar e Thandeewaran (2024)	Experimento empírico prático de monitoramento comportamental diante de vídeos e materiais adaptados.	Protocolos de análise de dados comportamentais voltados a aferir os indicadores de retenção de conteúdo e compreensão dos conceitos.
Yi et al. (2017)	Validação estatística do ganho de aprendizagem alcançado pelos alunos ao longo da trilha adaptativa.	Comparação de desempenho via exames empíricos (Pré-teste e Pós-teste) validados através de testes de significância estatística.

De modo geral, ocorre a ausência de discussões mais aprofundadas sobre barreiras e efeitos negativos, em que os benefícios são priorizados em detrimento de uma análise crítica mais equilibrada. Isso indica a necessidade de pesquisas futuras que, além de evidenciar ganhos, também aprofundem os desafios técnicos, pedagógicos e contextuais de implementação. Ainda assim, os resultados apontam para o potencial significativo dos sistemas adaptativos na promoção da aprendizagem em lógica de programação, reforçando seu papel como ferramenta inovadora no processo de ensino.



4. Conclusão

A presente RSL permitiu consolidar o estado da arte dos SAA aplicados ao ensino de LP, evidenciando contribuições voltadas à personalização e ao engajamento estudantil. Contudo, os achados revelam lacunas metodológicas e conceituais importantes que devem orientar a agenda de pesquisas futuras. Observou-se uma necessidade de ampliar o escopo de aplicação dessas tecnologias para além do ambiente universitário, estimulando investigações direcionadas à educação infantil, ao ensino médio e a contextos informais de aprendizagem, os quais ainda carecem de soluções adaptativas estruturadas. Além disso, o campo se beneficiará fortemente do desenvolvimento de estudos longitudinais e de discussões mais robustas sobre as barreiras técnicas e os impactos negativos dos SAA, permitindo uma compreensão aprofundada dos efeitos dessa personalização contínua ao longo do tempo.

Paralelamente, o direcionamento de novos trabalhos deve privilegiar metodologias de *design* participativo, promovendo o envolvimento ativo e contínuo do corpo docente desde as etapas iniciais de desenvolvimento das plataformas. Essa colaboração é fundamental para garantir que a lógica algorítmica de recomendação esteja intimamente alinhada às reais necessidades pedagógicas e à mediação humana em sala de aula. Outro ponto crítico diz respeito à forte dependência teórica de alguns SSA em Modelos de Estilos de Aprendizagem como pilares de customização. Os estudos analisados negligenciam o amplo debate científico contemporâneo liderado pela psicologia e pelas ciências da aprendizagem, que aponta para a escassez de evidências empíricas e estatísticas que sustentem a eficácia de adaptar conteúdos a perfis cognitivos estanques (UNIVERSITY OF MICHIGAN, 2025).

Portanto, para que os sistemas adaptativos avancem de forma ética e pedagogicamente sólida no ensino de programação, investigações futuras precisam estudar e analisar a visão de rotulagem dos estudantes, a qual pode induzir a crenças limitantes e afastar o aluno de múltiplos estímulos cognitivos. O amadurecimento tecnológico dos SAA deve, assim, convergir com abordagens cientificamente validadas de aprendizagem, integrando inteligência algorítmica, desenho instrucional robusto e a indispensável sensibilidade pedagógica do professor.

Referências

- ADNAN, M.; MUSSADIQ, S.; ASHRAF, J.; HABIB, A. Cloud-supported machine learning system for context-aware adaptive M-learning. *Turkish Journal of Electrical Engineering and Computer Sciences*, v. 27, n. 4, p. 2798-2816, 2019.
- ANITHA, D.; DEISY, C. Proposing a novel approach for classification and sequencing of learning objects in e-learning systems based on learning style. *Journal of Intelligent & Fuzzy Systems*, v. 29, n. 2, p. 539-552, 2015.
- BARBOSA, P. L. S.; CARMO, R. A. F. D.; GOMES, J. P.; VIANA, W. Adaptive learning in computer science education: A scoping review. *Education and Information Technologies*, v. 29, n. 8, p. 9139-9188, 2024.
- CHRYSAFIADI, K.; VIRVOU, M. Dynamically personalized e-training in computer programming and the language C. *IEEE Transactions on Education*, v. 56, n. 4, 2013.
- CORRÊA, D. G. M.; OLIVEIRA, T.; STRINGHINI, D. Estratégias focadas no ensino de lógica de programação: uma revisão sistemática da literatura brasileira. *In: Estudos Interdisciplinares nas Ciências Exatas e da Terra e Engenharias 3*. 1. ed. Ponta Grossa: Atena Editora, 2019.
- DOGAN, B.; DIKBIYIK, E. OPCOMITS: Developing an adaptive and intelligent web-based educational system based on concept map model. *Computer Applications in Engineering Education*, v. 24, n. 5, p. 676-691, 2016.



EVALE, D. S. Learning Management System with Prediction Model and Course-content Recommendation Module. *Journal of Information Technology Education: Research*, v. 16, n. 1, 2017.

GAVRILOVIĆ, N.; ARSIĆ, A.; DOMAZET, D.; MISHRA, A. Algorithm for adaptive learning process and improving learners' skills in Java programming language. *Computer Applications in Engineering Education*, v. 26, n. 5, p. 1362-1382, 2018.

HOED, R M. Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de Computação. Dissertação (Mestrado Profissional em Computação Aplicada) – Instituto de Ciências Exatas, Universidade de Brasília, Brasília, 2016.

KITCHENHAM, B. Procedures for Performing Systematic Reviews. Keele: Keele University, 2004. (Technical Report TR/SE-0401).

KUMAR, T. S. S.; THANDEESWARAN, R. Adapting video-based programming instruction: An empirical study using a decision tree learning model. *Education and Information Technologies*, v. 29, n. 11, p. 14205-14243, 2024.

LUCENA, T. B. S.; GOMES, P. S.; MUSA, D. L.; OLIVEIRA, T. Análise da aplicação de sistemas de recomendação no contexto educacional: revisão sistemática da literatura nacional. *RENOTE. Revista Novas Tecnologias na Educação*, v. 19, n. 2, p. 163-172, 2021.

MACENA, J.; PIRES, F.; PESSOA, M. Operação Lovelace: uma abordagem lúdica para introdução de aprendizagem em algoritmos. In: *SIMPÓSIO BRASILEIRO DE JOGOS E ENTRETENIMENTO DIGITAL (SBGames)*, 2020.

NABIZADEH, A. H.; JORGE, A. M.; LEAL, J. P. Estimating time and score uncertainty in generating successful learning paths under time constraints. *Expert Systems*, v. 36, n. 2, 2018.

NETO, J. G.; MARCOLINO, P. H. P.; FERREIRA, P.; BARBOSA, D. M. Maze code: Retórica procedural aplicada ao ensino de lógica de programação. In: *SIMPÓSIO BRASILEIRO DE JOGOS E ENTRETENIMENTO DIGITAL (SBGames)*, 2021.

SAITO, T.; WATANOBÉ, Y. Learning path recommender system based on recurrent neural network. In: *INTERNATIONAL CONFERENCE ON AWARENESS SCIENCE AND TECHNOLOGY (iCAST)*, 2018.

SANTOS, I. M. M.; KAVALERSKI, J. S. M. A. H.; DE SOUZA, T. J. G. As Aventuras Espaciais de Cody: protótipo de jogo para auxiliar no ensino de lógica de programação. In: *SIMPÓSIO BRASILEIRO DE JOGOS E ENTRETENIMENTO DIGITAL (SBGames)*, 2018.

SOGY, M.; KAUR, H.; GARG, A.; KHUNGER, M.; SEHGAL, R. Assessing Dynamic Learners' Behaviour Using Open Ended Tasks. In: *INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS AND COMPUTER NETWORKS (ISCON)*, 2019.

SOSNOVSKY, S.; BRUSILOVSKY, P. Evaluation of topic-based adaptation and student modeling in QuizGuide. *User Modeling and User-Adapted Interaction*, v. 25, n. 4, 2015.

UNIVERSITY OF MICHIGAN. The Myth of Learning Styles. Online Teaching, 2025. Disponível em: <https://onlineteaching.umich.edu/articles/the-myth-of-learning-styles/>. Acesso em: 31 maio 2026.

YI, B.; LV, Y.; LIU, H.; ZHANG, Z.; ZHANG, D.; WANG, Y. Lscb-als: a new als and its application in java language learning. In: *INTERNATIONAL SYMPOSIUM ON EDUCATIONAL TECHNOLOGY (ISET)*, 2017.