

Mapeando a aprendizagem em programação: Engajamento, desempenho e estruturas pedagógicas fundamentadas em Bloom e Kolb

Natália Caroline de Oliveira Rodrigues¹, Thiago Ayolphi Liuth²,
Andrey Braga de Lima², Javier Ulises Solis Lastra³,
Pedro Henrique Lopes Silva^{1,2}, Reinaldo Silva Fortes^{1,2}

¹ Programa de Pós-Graduação em Ciência da Computação, UFOP.

² Departamento de Computação, Universidade Federal de Ouro Preto (UFOP).

³ Escola Politécnica, Universidade de São Paulo (USP)

{natalia.cor, thiago.liuth, andrey.braga}@aluno.ufop.edu.br

jsolis@usp.br, {silvap, reifortes}@ufop.edu.br

Abstract. *Introductory programming education faces high dropout rates. This study analyzes pedagogical resources, teaching tools, and undergraduate student performance. Grounded in Bloom's and Kolb's frameworks, a five-step pedagogical model was developed and implemented over three semesters. We analyzed student data, interventions, dropout rates, and absenteeism. Performance remained heterogeneous across courses and unbalanced relative to Kolb's learning styles, with the Divergent profile at greater risk. While the volume of interactions and submissions is associated with success, uniform grades on the online judge system may not indicate actual knowledge acquisition. The paper concludes with a synthesis of lessons learned.*

Resumo. *O ensino introdutório de programação enfrenta altas taxas de evasão. Este estudo analisa recursos pedagógicos, ferramentas de ensino e o desempenho de estudantes de graduação. Com base nos referenciais de Bloom e Kolb, um modelo pedagógico de cinco etapas foi desenvolvido e aplicado ao longo de três semestres. Analisamos os dados dos estudantes, as intervenções, a evasão e as faltas. O desempenho permaneceu heterogêneo entre os cursos e desequilibrado em relação aos estilos de aprendizagem de Kolb, com o perfil Divergente em maior risco. Embora o volume de interações e submissões esteja associado ao sucesso, notas uniformes no online judge podem não indicar a real aquisição de conhecimento. O artigo conclui com uma síntese de lições aprendidas.*

1. Introdução

O ensino introdutório de programação em cursos de Ciências Exatas é frequentemente associado a dificuldades, desmotivação e evasão entre os discentes. Na Universidade Federal de Ouro Preto (UFOP), a disciplina “Programação de Computadores I” (BCC701) é oferecida em períodos iniciais de diferentes cursos, incluindo Engenharias, Matemática, Estatística e Química. As ofertas da disciplina respeitam um processo de unificação, visando promover maior equidade nos conteúdos e nos critérios de avaliação [Couch et al. 2023].

Embora a disciplina possua elementos unificados, como livro-texto, atividades práticas e avaliações teóricas, outros recursos, como o *Moodle*, videoaulas e o *opCoders Judge*, ainda são utilizados de forma heterogênea entre turmas e docentes. O *opCoders Judge*, é uma ferramenta para a correção automática das atividades práticas [Brito et al. 2019], mas seu uso, assim como o de outros recursos, precisa ser analisado de forma sistemática para compreender sua relação com o perfil dos discentes.

Nesse contexto, a principal lacuna reside na ausência de evidências consolidadas sobre como diferentes estratégias pedagógicas e padrões de uso dos recursos educacionais se associam ao sucesso ou ao insucesso acadêmico na disciplina. Essa lacuna é particularmente relevante porque muitos docentes de programação possuem formação predominantemente técnica e nem sempre dispõem de formação pedagógica específica. Como consequência, o ensino pode priorizar a dimensão técnica da programação, sem explicitar adequadamente os processos pelos quais os discentes constroem seu conhecimento.

Sendo assim, esta pesquisa tem como objetivo analisar a relação entre dados de acesso e uso de objetos de aprendizagem, engajamento discente e desempenho acadêmico na disciplina BCC701. Para isso, serão considerados registros provenientes do *Moodle*, do *opCoders Judge* e de atividades de suporte, como tutorias e debates *online*. A análise busca identificar padrões de comportamento associados à aprovação, à retenção e às dificuldades de aprendizagem, oferecendo subsídios para o aprimoramento das estratégias pedagógicas aplicadas na disciplina.

A investigação fundamenta-se em três referenciais. A *Taxonomia de Bloom* é utilizada para apoiar a organização dos objetivos de aprendizagem e a progressão cognitiva das atividades, variando de níveis básicos de compreensão a tarefas que envolvam análise, avaliação e criação [dos Santos et al. 2026]. O *Ciclo de Aprendizagem de Kolb* orienta a análise da aprendizagem como um processo que articula *experiência concreta*, *reflexão*, *conceitualização* e *aplicação* [Kolb 2015]. Complementarmente, o *Inventário de Estilos de Aprendizagem* é empregado para mapear as preferências dos discentes e apoiar a proposição de estratégias mais alinhadas a diferentes perfis de aprendizagem [Kolb 2015].

Como principais contribuições, este estudo visa fornecer subsídios teóricos e práticos para a formulação de diretrizes pedagógicas no ensino inicial de programação em Ciências Exatas. Os achados visam orientar a escolha e a incorporação de recursos didáticos de forma ampla, além de fundamentar a atuação docente e viabilizar a personalização do aprendizado, atuando estrategicamente na redução da retenção e da evasão nesses cenários. Mais especificamente, listamos as principais contribuições do trabalho: (i) Realização de intervenções graduais ao longo dos últimos 3 semestres, com mapeamento dos objetos de aprendizagem ao Ciclo de Kolb e alcançando cobertura de todas as suas fases; (ii) Elaboração de um modelo pedagógico para o ensino de programação introdutória à luz da teoria da aprendizagem experiencial de Kolb e de forma articulada ao Ciclo de Kolb; (iii) Uma análise exploratória de resultados de desempenho acadêmico dos discentes e sua relação com os objetos de aprendizado da disciplina e estilos de aprendizagem de Kolb; e (iv) uma síntese de lições aprendidas.

O restante deste trabalho é dividido da seguinte forma. A Seção 2 discute trabalhos relacionados e a Seção 3 apresenta o desenvolvimento metodológico da disciplina BCC701. Posteriormente, a Seção 4 apresenta uma análise exploratória do desempenho

acadêmico dos discentes, enquanto a Seção 5 apresenta uma síntese das lições aprendidas a partir destas análises e a Seção 6 conclui o trabalho.

2. Trabalhos Relacionados

O ensino introdutório de programação no ensino superior tem sido investigado sob diferentes perspectivas pedagógicas e tecnológicas, especialmente em contextos em que discentes enfrentam dificuldades associadas à abstração conceitual, à lógica algorítmica e à aplicação prática de conceitos [de Sousa Oliveira et al. 2024]. Estudos recentes indicam que abordagens centradas apenas na exposição teórica tendem a ser insuficientes para sustentar o engajamento e a progressão da aprendizagem, reforçando a necessidade de práticas orientadas à resolução de problemas, projetos reais e uso de recursos tecnológicos [de Sousa Oliveira et al. 2024]. Nesse cenário, ambientes virtuais de aprendizagem, corretores automáticos, metodologias ativas e estratégias de personalização têm sido empregados para ampliar as oportunidades de prática, oferecer acompanhamento contínuo e apoiar intervenções pedagógicas baseadas em evidências [Lima et al. 2024, Messer et al. 2024, Pereira et al. 2024, Wu and Wan 2025].

Ambientes virtuais de aprendizagem, como o *Moodle*, são frequentemente utilizados não apenas como repositórios de conteúdo, mas também como ecossistemas de mediação pedagógica, interação e monitoramento discente, permitindo acompanhar padrões de participação, identificar comportamentos de risco e apoiar decisões docentes [Lima et al. 2024, Porto et al. 2023, Carvalho et al. 2025]. Em disciplinas introdutórias de programação, esses dados podem ser particularmente relevantes para compreender a relação entre engajamento, assiduidade nas atividades práticas e desempenho acadêmico [Lima et al. 2024]. Complementarmente, corretores automáticos de código, como o *op-Coders Judge*, ampliam a escalabilidade da avaliação ao fornecer *feedback* imediato sobre submissões de programas [Messer et al. 2024, da Silva et al. 2024]. Contudo, a literatura também aponta limitações recorrentes dessas ferramentas, sobretudo quando a avaliação se restringe à correteza funcional e oferece pouco suporte qualitativo para a compreensão dos erros e para a melhoria do código [Messer et al. 2024].

Outra linha de trabalhos enfatiza metodologias ativas e aprendizagem experiencial como estratégias para tornar o ensino de programação mais significativo [Kolb 2015, Konak et al. 2014, Taylor 2007, Botelho et al. 2016, Marietto et al. 2014]. Estudos baseados no *Ciclo de Kolb* destacam a importância de articular experiência concreta, reflexão, conceitualização e experimentação ativa, especialmente em atividades relacionadas ao ensino de programação [Kolb 2015, Konak et al. 2014, Govender and Govender 2023, Akkaya and Akpinar 2022, de Souza et al. 2024]. Essas abordagens deslocam o discente de uma posição passiva para uma atuação mais participativa, na qual o erro, a análise de soluções e a reformulação de estratégias tornam-se parte do processo formativo [Kolb 2015, Konak et al. 2014, Taylor 2007]. De modo complementar, a *Taxonomia de Bloom* tem sido utilizada para orientar a progressão cognitiva das atividades, permitindo alinhar objetivos, práticas e avaliações em níveis que vão da compreensão inicial à análise, avaliação e criação de soluções computacionais [Sobral 2021, Widian et al. 2023, Tamborlin and Piton-Gonçalves, Garcia et al. 2023].

Trabalhos recentes também investigam a personalização e a inteligência artificial no ensino de programação. Modelos adaptativos, tutores inteligentes, bancos de questões

personalizadas, visualizações dinâmicas de algoritmos e ferramentas de IA generativa têm sido propostos para oferecer suporte individualizado, *feedback* contextualizado e maior interação com o conteúdo [Li et al. 2025, Sanal Kumar and Thandeeswaran 2024, Wu and Wan 2025, Chang et al. 2024]. Embora esses estudos indiquem benefícios potenciais para engajamento, compreensão e autonomia, ainda há desafios relacionados à integração pedagógica dessas tecnologias, à qualidade das evidências empíricas e à articulação entre dados de uso, desempenho acadêmico e desenho instrucional [Maia et al. 2017, Masegosa et al. 2024, Chang et al. 2024]. Além disso, pesquisas sobre fatores socioculturais mostram que a aprendizagem de programação não envolve apenas domínio técnico, mas também pertencimento, colaboração, autoconfiança e reconhecimento do discente como sujeito capaz de atuar na área [Johnson Jr et al. 2025, Tzafilkou et al. 2020, Choi and Yang 2024, Luo and Adelopo 2025].

Em síntese, a literatura aponta avanços relevantes no uso de ambientes virtuais, avaliação automática, metodologias ativas, aprendizagem experiencial, *Learning Analytics*, personalização e IA no ensino de programação. No entanto, muitos estudos permanecem concentrados em intervenções específicas ou em contextos isolados, com menor ênfase na análise integrada e longitudinal entre engajamento, desempenho, organização didática e progressão cognitiva. Esta pesquisa se posiciona nessa lacuna ao articular dados retrospectivos e prospectivos da disciplina BCC701, considerando registros do *Moodle*, submissões no *opCoders Judge* e desempenho acadêmico, com o objetivo de compreender padrões de aprendizagem e subsidiar práticas pedagógicas mais alinhadas à aprendizagem experiencial e à *Taxonomia de Bloom*.

3. Desenvolvimento metodológico da intervenção

Esta seção descreve o desenvolvimento metodológico da intervenção realizada na disciplina BCC701. A disciplina é ofertada a um grande número de discentes, em mais de 20 turmas semestrais, envolvendo diversos docentes. Neste contexto, ao longo dos últimos anos, foi construída uma estrutura unificada, composta por um plano de ensino, um livro-texto digital, materiais de apoio, atividades práticas e avaliações teóricas comuns. Embora essa estrutura seja compartilhada entre as turmas, sua implementação varia conforme o docente, especialmente no uso do *Moodle*, do *opCoders Judge* e de estratégias ativas.

Este estudo foca na organização mais recente dos recursos didáticos, implementada entre os semestres 2024/2, 2025/1 e 2025/2. As mudanças foram definidas e analisadas à luz da *Teoria da Aprendizagem Experiencial de Kolb* [Kolb 2015] e da *Taxonomia de Bloom* [Widiana et al. 2023], buscando compreender como os variados recursos da disciplina podem favorecer diferentes etapas do processo de aprendizagem e os estilos de aprendizagem dos discentes. Consideramos as turmas em que o *Moodle* e o *opCoders Judge* foram utilizados de forma integrada, permitindo ao discente acessar materiais, resolver atividades e submeter seus programas à correção automática com *feedback*.

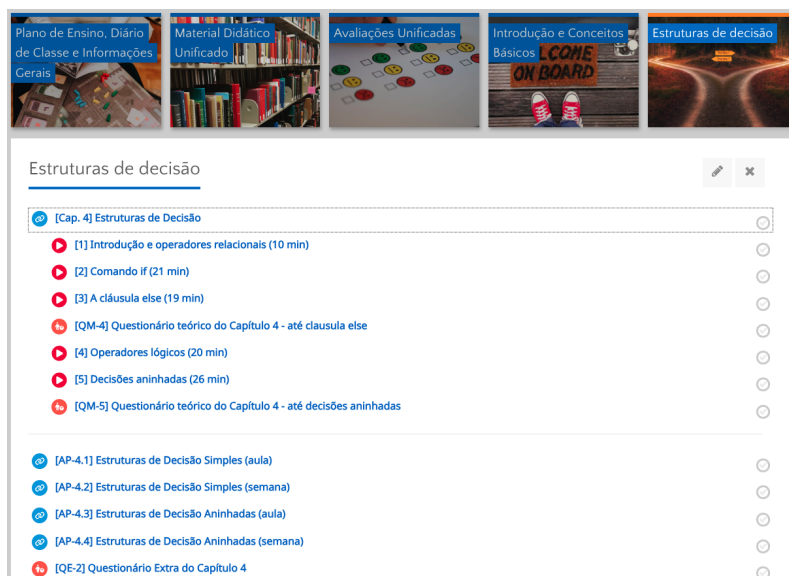
A seguir, a Seção 3.1 apresenta as intervenções realizadas e a organização dos recursos didáticos. Posteriormente, a Seção 3.2 apresenta o modelo pedagógico proposto para a exploração dos recursos didáticos da disciplina.

3.1. Intervenções e organização dos recursos didáticos

As intervenções foram propostas de forma incremental ao longo dos últimos semestres. O *Moodle* centraliza o acesso a todas as informações e recursos da disciplina, concentrando

as intervenções propostas. A Figura 1 apresenta uma visão geral da organização final do *Moodle*. Os recursos são organizados em blocos, cada módulo possui recursos relacionados às aulas teóricas, questionários teóricos, atividades práticas e questionários extras. As intervenções propostas e detalhamento sobre os recursos são descritos a seguir. Neste processo, associamos cada recurso ao Ciclo de Kolb, visando a uma compreensão mais aprofundada deles segundo preceitos pedagógicos.

Figura 1. Visão geral do *Moodle* após intervenções.



O semestre 2024/2 foi adotado como cenário base. Nesse período, a disciplina era organizada principalmente em torno da apresentação conceitual dos conteúdos, por meio de videoaulas, *slides*, livro-texto, resumos e atividades práticas. Essa configuração favorecia sobretudo a *conceitualização abstrata*, enquanto a *experiência concreta* aparecia de forma menos explícita, geralmente após a exposição teórica. As videoaulas passaram a ter duração menor, de aproximadamente 30 minutos, em substituição a vídeos mais longos dos semestres anteriores. Essa alteração buscou favorecer o engajamento discente com objetos de aprendizagem mais curtos, em consonância com [López-Pernas et al. 2021]. O bloco “*Informações Gerais*”, reúne plano de ensino, cronograma e informações administrativas. O bloco “*Material Didático Unificado*”, reúne livro-texto, resumos das avaliações e bibliografia. Nos demais blocos, organizam-se os recursos de cada módulo do conteúdo programático da disciplina, compostos por *slides*, videoaulas e atividades práticas em PDF, também disponíveis no *opCoders Judge*. Nessa organização, os materiais teóricos foram associados à *conceitualização abstrata*, as atividades práticas à *experimentação ativa* e o *feedback* do *opCoders Judge* à *observação reflexiva*.

Em 2025/1, foram introduzidas três reformulações principais articuladas à dinâmica do Ciclo de Aprendizagem Experiencial de Kolb: (i) a segmentação das videoaulas em subtópicos, que ampara a Conceitualização Abstrata ao reduzir a sobrecarga cognitiva e facilitar a revisão seletiva; (ii) a inclusão de questionários teóricos pós-módulo como instrumentos de autorregulação, estimulando a Observação Reflexiva para a identificação de lacunas conceituais; e (iii) a reorganização das Atividades Práticas, que fecha o ciclo de Kolb. Nesta última, a etapa guiada em aula proporciona a Experiência

Concreta do desenvolvimento de código em ambiente compartilhado, enquanto a etapa semanal volta-se à resolução autônoma de problemas, impulsionando a Experimentação Ativa ao desafiar o discente a aplicar o conhecimento de forma mais independente.

Em 2025/2, a intervenção foi ampliada com a inclusão do *fórum de dúvidas*, do *questionário pré-curso*, do bloco de “*Avaliações Unificadas*” e do “*Questionário Extra*”. O fórum foi proposto como um espaço de interação entre docentes e discentes, favorecendo a formulação de dúvidas, a troca de explicações e a reflexão coletiva. O questionário pré-curso foi utilizado para identificar conhecimentos prévios e estabelecer um ponto de partida para o acompanhamento da aprendizagem. Um novo bloco, “*Avaliações Unificadas*”, foi incluído para centralizar informações sobre datas, procedimentos, gabaritos e critérios de correção das avaliações. Esse recurso foi associado à *observação reflexiva*, por apoiar a revisão dos resultados e a compreensão dos critérios de avaliação. Também foi proposta uma mudança na abordagem inicial da disciplina, estimulando os discentes a resolver problemas de alto nível, por meio de algoritmos e fluxogramas, antes da implementação em código. Essa estratégia buscou fortalecer a *experiência concreta*, ao aproximar o discente do problema antes da formalização teórica. Por fim, foi incluído o recurso “*Questionário Extra*”, composto por questões discursivas sobre as dúvidas dos discentes e a proposição de novos exercícios práticos. Essa intervenção foi associada à *observação reflexiva* e à *conceituação abstrata*.

Com base na síntese apresentada na Tabela 1, a inserção e a reestruturação dos recursos didáticos ao longo dos semestres constituíram intervenções pedagógicas fundamentadas nas teorias de Kolb e Bloom. O *Ciclo de Kolb* contribuiu para a diversificação dos materiais e para a vivência da aprendizagem experiencial. Paralelamente, essas intervenções foram estruturadas com base na *Taxonomia de Bloom* para garantir uma progressão cognitiva adequada. Os recursos teóricos e os questionários iniciais foram organizados para consolidar habilidades básicas (*lembrar e compreender*).

Tabela 1. Associações analíticas entre recursos didáticos da disciplina e sua associação às etapas do *Ciclo de Kolb*.

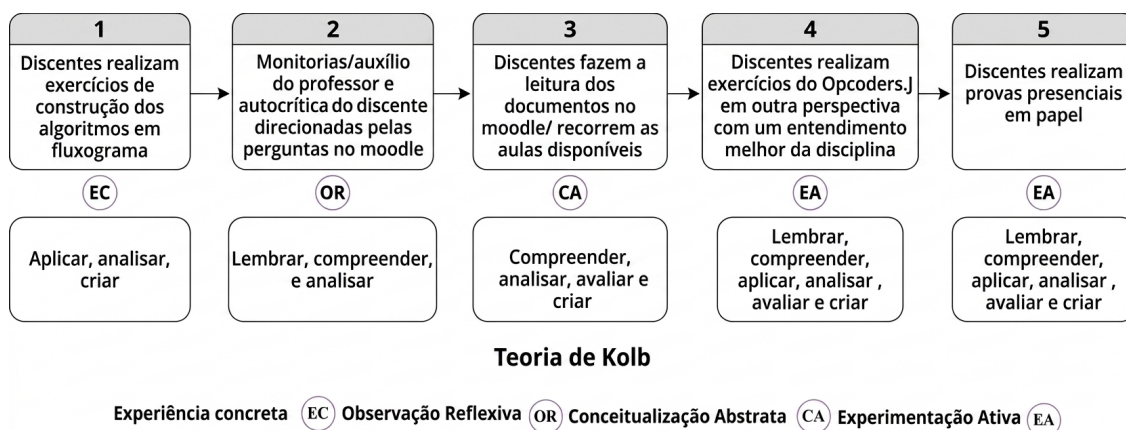
Semestre	Recurso ou estrutura	Associação com o <i>Ciclo de Kolb</i>
2024/2	Livro-texto e resumos	<i>Conceituação abstrata</i> : contato com conceitos, fundamentos e referências sintáticas.
	Videoaulas e <i>slides</i>	<i>Conceituação abstrata</i> : apresentação estruturada dos conteúdos.
	Atividades no <i>opCoders Judge</i>	<i>Experimentação ativa</i> : aplicação de conceitos em problemas de programação.
	<i>Feedback</i> do <i>opCoders Judge</i>	<i>Observação reflexiva</i> : análise de erros, testes e resultados.
2025/1	Videoaulas segmentadas	<i>Conceituação abstrata</i> : estudo dos conteúdos em unidades menores.
	Questionários teóricos	<i>Observação reflexiva</i> : verificação da compreensão e identificação de lacunas.
	Atividades Práticas em etapas guiada e autônoma	<i>Experiência concreta</i> e <i>experimentação ativa</i> : vivência da resolução e aplicação de conceitos.
2025/2	Fórum de dúvidas	<i>Observação reflexiva</i> e <i>conceituação abstrata</i> : troca de raciocínios e construção coletiva de significado.
	Questionário pré-curso	<i>Experiência concreta</i> : reconhecimento do ponto de partida.
	Bloco de avaliações unificadas	<i>Observação reflexiva</i> , <i>conceituação abstrata</i> e <i>experimentação ativa</i> : revisão de resultados, critérios e preparação.
	Questionário Extra	<i>Observação reflexiva</i> e <i>conceituação abstrata</i> : análise de dificuldades e formulação de problemas.

3.2. Modelo pedagógico proposto

Com base nas intervenções descritas anteriormente, a disciplina foi compreendida como um ambiente de aprendizagem experiencial, no qual os recursos digitais e presenciais são organizados para favorecer momentos de *experiência*, *reflexão*, *conceituação* e *experimentação*. Essa organização considera que diferentes estratégias didáticas podem favorecer diferentes formas de envolvimento com o conhecimento, conforme a *Teoria da Aprendizagem Experiencial de Kolb* [Kolb 2015].

O modelo proposto articula o *Ciclo de Kolb* à *Taxonomia de Bloom*. Inicialmente, os discentes são estimulados a lidar com problemas por meio de algoritmos e fluxogramas; em seguida, acessam materiais teóricos no *Moodle*; posteriormente, recebem apoio por meio de aulas, monitorias, tutorias e *feedback* do *opCoders Judge*; por fim, retomam os exercícios e realizam avaliações presenciais, mobilizando diferentes níveis cognitivos. A Figura 2 sintetiza essa proposta, elaborada com base em [Solis-Lastra et al. 2024] e [Ferraz and Belhot 2010].

Figura 2. Modelo pedagógico proposto para a disciplina. Baseado em [Solis-Lastra et al. 2024] e [Ferraz and Belhot 2010].



O modelo ilustra a arquitetura pedagógica da disciplina por meio de um roteiro de cinco etapas, as quais representam o *Ciclo da Aprendizagem Experiencial de Kolb* e a progressão cognitiva da *Taxonomia de Bloom*. O fluxo inicia-se com um desafio prático de algoritmos em fluxogramas (*Experiência Concreta*), o aprendizado é depurado por meio de autocrítica e de interação com a docência (*Observação Reflexiva*), seguido da busca de fundamentação teórica nos materiais estruturados do curso (*Conceitualização Abstrata*), culminando no retorno à prática avançada no sistema e nas avaliações somativas presenciais (*Experimentação Ativa*). Ao longo de toda essa trajetória experiencial, as demandas cognitivas evoluem gradativamente, exigindo desde a consolidação de habilidades de base até a mobilização de todos os níveis da *Taxonomia de Bloom* na aplicação, na análise e na criação de soluções.

4. Análise exploratória de resultados

Esta seção apresenta uma análise exploratória do desempenho acadêmico dos discentes e de sua interação com os recursos digitais da disciplina. Foram considerados registros acadêmicos, notas, frequência, submissões no *opCoders Judge*, acessos ao *Moodle* e respostas a questionários. As análises concentram-se nos semestres 24/2, 25/1 e 25/2 (este

último encontrava-se em andamento no momento da realização desta pesquisa e, portanto, não consta em todas as análises). Os estilos de aprendizagem foram analisados segundo a tipologia de Kolb [Kolb 2015]. O objetivo desta análise é identificar padrões e caracterizar o engajamento e o desempenho dos discentes ao longo dos semestres com intervenções, bem como analisar a relação desses aspectos com os estilos de aprendizagem de Kolb.

A Seção 4.1 apresenta uma caracterização do desempenho acadêmico dos discentes, enquanto as Seções 4.2 e 4.3 apresentam caracterizações do uso das ferramentas *opCoders Judge* e *Moodle*, respectivamente. Em seguida, na Seção 4.4 é apresentada uma caracterização dos estilos de aprendizagem de Kolb.

4.1. Caracterização de desempenho acadêmico

A Tabela 2 apresenta os indicadores acadêmicos dos semestres 24/2 e 25/1. Entre os dois semestres, houve redução das desistências, de 26,7% para 19,0%, e das reprovações por nota e falta, de 22,4% para 12,7%. Também houve aumento de concluintes e aprovados. Entretanto, a reprovação por nota permaneceu estável, em torno de 31% dos concluintes, indicando que parte expressiva dos discentes permanece até o final da disciplina, mas não alcança desempenho suficiente para aprovação. Embora o escopo atual não permita a validação definitiva das proposições neste momento, esse panorama atua como um diagnóstico inicial que fundamenta trabalhos futuros.

Tabela 2. Quantitativos de discentes matriculados nos semestres 24/2 e 25/1.

	24/2	25/1
# Matrículas	499	525
# Desistentes	133 (26,7%)	100 (19,0%)
# Concluintes	366 (73,3%)	425 (81,0%)
# Aprovados	170 (46,4%)	238 (56,0%)
# Aprovados com Exame	42 (11,5%)	60 (14,1%)
# Reprovados por Nota	114 (31,1%)	133 (31,3%)
# Reprovados por Nota e Falta	82 (22,4%)	54 (12,7%)

A Figura 3 apresenta uma análise do resultado de aprovação por curso, indicando diferenças entre grupos, sobretudo nas Engenharias, que concentram a maior parte das matrículas. Engenharia Civil, Controle e Automação, Produção e Mecânica apresentaram maior proporção de aprovações, enquanto Engenharia Ambiental, Minas, Metalúrgica, Urbana e Química Industrial concentraram maior proporção relativa de reprovações. Esses resultados demonstram que as estratégias pedagógicas e os recursos didáticos precisam ser diversificados, evidenciando que a adoção de recursos personalizados ou adaptativos pode ser essencial para a melhoria dos resultados gerais da disciplina.

A Figura 4 apresenta diagramas de caixa das notas finais e das faltas dos discentes. A comparação entre notas e faltas mostra que os aprovados apresentam nota final muito superior à dos reprovados, e uma relação inversa em relação ao percentual de faltas. Assim, a reprovação pode estar diretamente associada à menor assiduidade. Portanto, iniciativas que incentivem a presença dos discentes nas aulas mostram-se relevantes para a melhoria dos resultados gerais da disciplina, bem como fomentar o acesso regular aos recursos assíncronos e *online* complementares, como livro-texto e videoaulas.

4.2. Caracterização do uso do *opCoders Judge*

O desempenho dos discentes no *opCoders Judge* foi analisado considerando todas as submissões e apenas as entregas contabilizadas para nota. A Figura 5 apresenta a distribuição

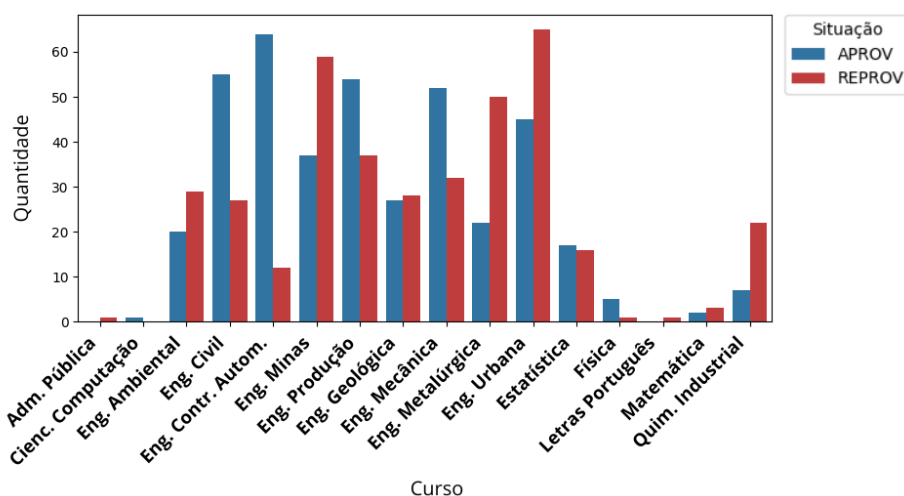
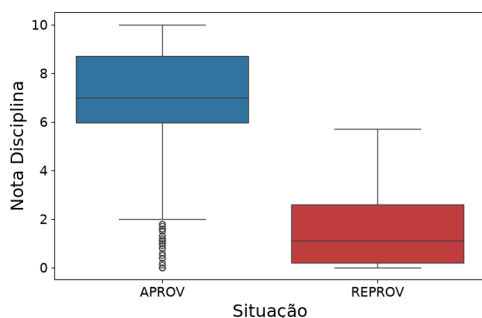
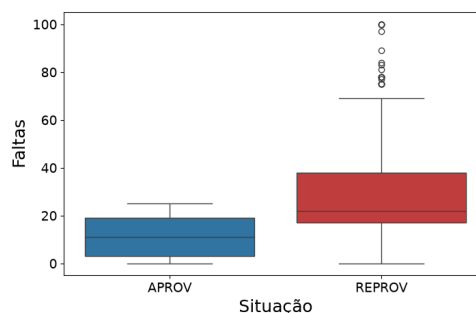


Figura 3. Resultados por curso, agregando 24/2 e 25/1.

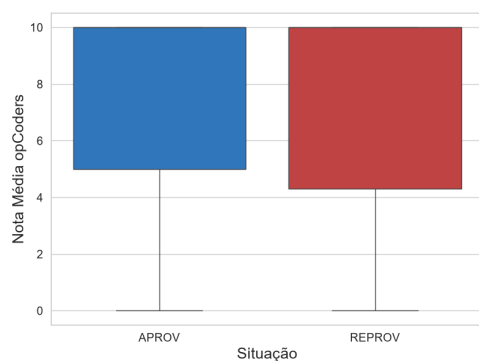


(a) Nota final.

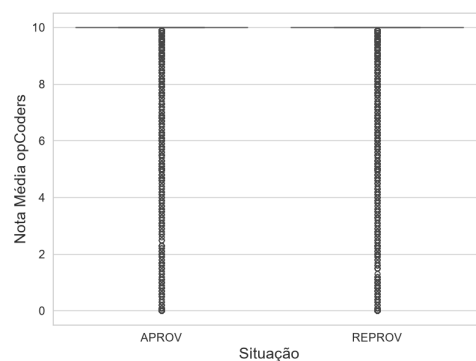


(b) Faltas.

Figura 4. Nota final e percentual de faltas, agregando 24/2 e 25/1.



(a) Todas as entregas.



(b) Entregas contabilizadas.

Figura 5. Distribuição das notas no *opCoders Judge* por situação acadêmica.

de notas obtidas pelos discentes por situação acadêmica. A nota média na plataforma foi ligeiramente maior entre os aprovados, mas a diferença em relação aos reprovados foi pequena, sobretudo nas entregas contabilizadas. Portanto, constata-se que obter boas notas no *opCoders Judge* não foi suficiente para garantir aprovação. Esses resultados indicam a necessidade de revisão do formato e do tipo de cobrança das atividades práticas realizadas por meio da plataforma.

Em contraste, o número de entregas diferencia melhor os grupos, conforme demonstram os gráficos da Figura 6. Os aprovados concentram-se em faixas mais altas de submissão, enquanto os reprovados apresentam maior concentração em baixos volumes de entregas. Assim, a regularidade de interação com a plataforma parece mais relevante do que a nota média isolada. O fato de haver um número menor de entregas entre os alunos reprovados, aliado a pequenas diferenças nas notas em relação aos aprovados, indica que a resolução de exercícios práticos tem ocorrido de forma ineficiente e precisa ser aprimorada. Esse cenário sugere que os discentes podem estar obtendo ajuda, lícita ou ilícita, seja por meio do uso de respostas prontas, seja por uma excessiva dependência de auxílio durante as aulas e atendimentos extraclasse.

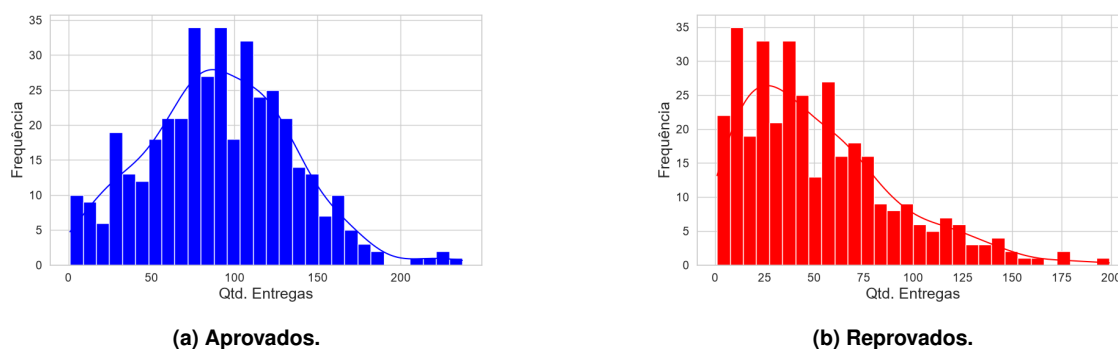


Figura 6. Número de entregas por discente, agregando 24/2 e 25/1.

4.3. Uso do Moodle

O uso do *Moodle* foi analisado a partir da presença ou ausência de data de acesso aos recursos disponibilizados. Observou-se um baixo volume de acesso aos recursos, com percentuais de acessos de 18,89% em 24/2 e 13,77% em 25/1. Porém, esse resultado possivelmente está relacionado ao caráter presencial da disciplina e ao uso do *opCoders Judge* nas atividades práticas.

A Figura 7 apresenta a relação entre a nota final e o percentual de acesso ao *Moodle*. A linha de tendência indica um resultado positivo para quem mais acessa os recursos. No entanto, há discentes com baixo acesso e notas elevadas, o que indica que o uso da plataforma não é condição obrigatória para um bom desempenho. Ainda assim, maior acesso parece estar associado a melhores notas na disciplina e à redução do risco de reprovação.

4.4. Caracterização dos estilos de aprendizagem de Kolb

Para avaliar o perfil dos discentes, foi aplicado um questionário com perguntas baseado no Inventário de Estilos de Aprendizagem de Kolb para colhermos o perfil de aprendizagem dos estudantes. A Tabela 3 apresenta dados quantitativos sobre a classificação dos discentes quanto aos estilos de aprendizagem e a Figura 8 apresenta indicadores acadêmicos segmentados por perfil. A distribuição dos estilos mostra predominância dos perfis *Acomodador* e *Divergente*. O perfil *Acomodador* apresentou os melhores indicadores acadêmicos, com maior nota, maior uso do *Moodle* e menor percentual de faltas. O perfil *Divergente* apresentou menor desempenho e maior percentual de faltas, o que configura um possível grupo de risco. Estes resultados indicam um desequilíbrio no atendimento aos diferentes perfis, apontando para a necessidade de novas intervenções que possibilitem um maior equilíbrio entre eles.

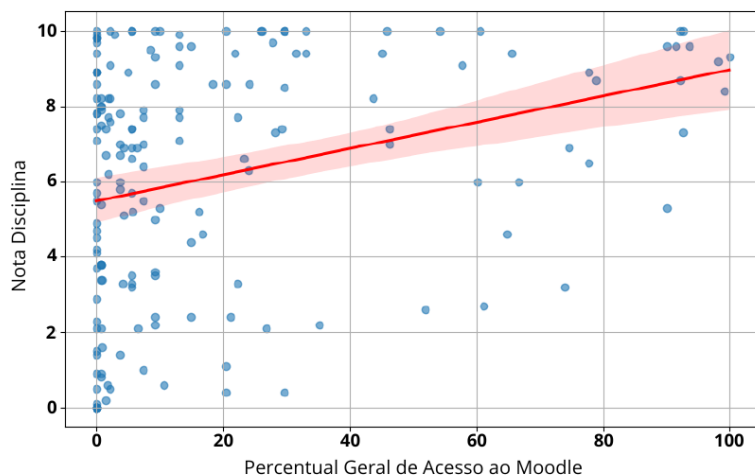


Figura 7. Relação entre nota final e percentual de acesso ao Moodle.

Tabela 3. Classificação dos discentes nos estilos de aprendizagem de Kolb.

	25/1	25/2	Total
Acomodador (EC/EA)	90	74	164
Assimilador (CA/OR)	28	31	59
Convergente (CA/EA)	30	29	59
Divergente (EC/OR)	60	65	125
Total	208	199	407

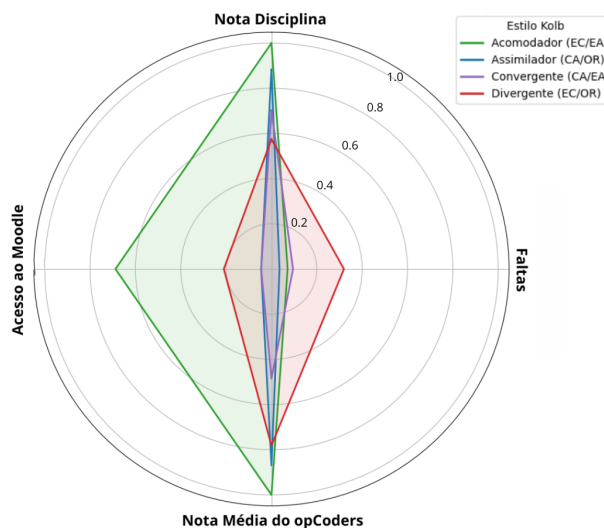


Figura 8. Indicadores acadêmicos por estilo de aprendizagem de Kolb.

5. Lições aprendidas

Esta seção apresenta uma síntese dos resultados como uma listagem de lições aprendidas:

1. O desempenho pode variar de acordo com o curso do discente, evidenciando que os objetos de aprendizado devem ser diversificados, personalizados e adaptativos;
2. O desempenho varia de acordo com o grau de assiduidade dos discentes, evidenciando a necessidade de incentivo à frequência em aulas e acesso aos objetos de aprendizado assíncronos e *online* complementares;
3. O uso do *opCoders Judge* não demonstrou evidências de ganho de desempenho em nota, embora seja observado que alunos aprovados tendem a fazer mais entregas de soluções, evidenciando a necessidade de revisão do formato e do tipo de cobrança das atividades práticas realizadas por meio de plataformas de autocorreção, bem como uma reavaliação do atendimento em aulas práticas e extraclasse;

4. Apesar do baixo acesso aos recursos do *Moodle*, observaram-se indícios de que o acesso regular contribui para a obtenção de melhores notas e para a redução do risco de reprovação;
5. Apesar dos esforços para oferecer variadas atividades avaliativas e recursos complementares às aulas presenciais, o desempenho entre os diferentes estilos de aprendizagem de Kolb ainda é desequilibrado, o que evidencia a necessidade de novas intervenções que promovam maior equilíbrio e melhorias no desempenho.

6. Conclusão

Este estudo apresentou uma análise do ensino introdutório de programação em cursos de Ciências Exatas, investigando as relações entre o engajamento dos discentes, os padrões de uso de recursos tecnológicos e o desempenho acadêmico sob a ótica combinada do Ciclo de Aprendizagem Experiencial de Kolb e da Taxonomia de Bloom.

Conclui-se que as intervenções incrementais aplicadas ao longo dos últimos semestres demonstraram eficácia na redução dos índices de evasão e de reprovações por falta, consolidando uma estrutura pedagógica unificada voltada à progressão cognitiva dos estudantes. Os resultados obtidos trouxeram lições valiosas e contraintuitivas para o desenho instrucional da disciplina. Embora a assiduidade e a constância nas entregas de exercícios práticos se associem fortemente ao sucesso acadêmico, a homogeneidade observada nas notas da plataforma de correção automática entre aprovados e reprovados sinaliza a necessidade de revisar o formato e a natureza das atividades práticas assíncronas, a fim de mitigar dependências excessivas ou atitudes ilícitas. Ademais, a identificação do perfil Divergente como grupo de risco acadêmico reforça a ideia de que as estratégias tradicionais de ensino falham em atender de forma equitativa aos diferentes estilos de aprendizagem.

Como limitações, reconhece-se o caráter preliminar e exploratório dos dados com dois períodos letivos concluídos e um período parcial sendo avaliados, o qual carece de análises estatísticas inferenciais definitivas. Para trabalhos futuros, planeja-se realizar a validação estatística rigorosa das intervenções propostas em mais períodos subsequentes, bem como o desenvolvimento e a implementação de trilhas de aprendizagem personalizadas e adaptativas, capazes de direcionar recursos específicos a cada estilo de aprendizagem e de atuar de forma preditiva no suporte a estudantes com dificuldades precoces.

Agradecimentos

Gostaríamos de agradecer às nossas respectivas instituições (UFOP e USP), à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) – códigos APQ-06466-24 e APQ-01768-24.

Cuidados éticos

Por envolver a análise de registros acadêmicos e respostas a questionários de participantes humanos (corpo discente), esta pesquisa foi conduzida em estrita observância aos preceitos éticos estabelecidos pelas Resoluções nº 466/2012 e nº 510/2016 do Conselho Nacional de Saúde (CNS). O projeto recebeu aprovação prévia do Comitê de Ética em Pesquisa (CEP), estando registrado sob o parecer técnico de aprovação

número 90515925.6.0000.5097. Garantiu-se o anonimato dos participantes, o sigilo das informações coletadas e o uso estrito dos dados para finalidades estritamente científicas, sem prejuízo ou exposição de qualquer ordem aos estudantes matriculados.

Declaração sobre uso de Inteligência Artificial

Em estrito atendimento ao Código de Conduta para autores da Sociedade Brasileira de Computação (SBC), declara-se explicitamente que ferramentas de Inteligência Artificial (IA) Generativa foram empregadas exclusivamente nas etapas de revisão gramatical, aprimoramento estilístico de conteúdo e refinamento de coesão textual de trechos deste artigo. Destaca-se que tais ferramentas computacionais não foram listadas e nem cumprem requisitos para autoria ou coautoria do trabalho. Os autores humanos assumem total e integral responsabilidade sobre todo o conteúdo intelectual e científico apresentado, garantindo a originalidade do texto e a ausência de plágio em todas as seções constituintes do artigo.

Referências

- Akkaya, A. and Akpinar, Y. (2022). Experiential serious-game design for development of knowledge of object-oriented programming and computational thinking skills. *Computer Science Education*, 32(4):476–501.
- Botelho, W. T., Marietto, M. d. G. B., Ferreira, J. C. d. M., and Pimentel, E. P. (2016). Kolb's experiential learning theory and belhot's learning cycle guiding the use of computer simulation in engineering education: A pedagogical proposal to shift toward an experiential pedagogy. *Computer Applications in Engineering Education*, 24(1):79–88.
- Brito, P. S. S., Fortes, R. S., and Faria, F. L. d. M. (2019). Programação competitiva como ferramenta de apoio ao ensino de algoritmos e estrutura de dados para alunos de ciência da computação. In *Anais do XXX Simpósio Brasileiro de Informática na Educação (SBIE 2019)*, pages 833–842, Porto Alegre, RS, Brasil. SBC.
- Carvalho, G., Silva, J. H. N. d. O., Carvalho, Á. M., Fonseca, Y. M. d. A., Beccari, J. B., Veloso, R. R., de Souza, J. F., and Barrere, E. (2025). Análise de aprendizagem no moodle: Construção de um painel de dados com indicadores educacionais. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 85–98. SBC.
- Chang, C. I., Choi, W. C., and Choi, I. C. (2024). A systematic literature review of the opportunities and advantages for aigc (openai chatgpt, copilot, codex) in programming course. In *Proceedings of the 2024 7th International Conference on Big Data and Education*, pages 29–35.
- Choi, J.-I. and Yang, S. (2024). Effectiveness and design of pbl-based project approach for non-major university computing courses. *Applied Sciences*, 15(1):50.
- Couch, B. A., Prevost, L. B., Stains, M., Whitt, B., Marcy, A. E., Apkarian, N., Dancy, M. H., Henderson, C., Johnson, E., Raker, J. R., Yik, B. J., Earl, B., Shadle, S. E., Skvoretz, J., and Ziker, J. P. (2023). Examining whether and how instructional coordination occurs within introductory undergraduate stem courses. *Frontiers in Education*, Volume 8 - 2023.

- da Silva, L. L., Cedraz, V. F., do Patrocínio, J. A., Delabrida, S., and Fortes, R. S. (2024). Juiz online focado no ensino de programação: uma análise de usabilidade de ferramenta autoral. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 1234–1247. SBC.
- de Sousa Oliveira, K. K., da Silva Marcolino, A., Falcão, T. P., and Barbosa, E. F. (2024). Ensino e aprendizagem de programação na educação 4.0: Um mapeamento sistemático da literatura. *Simpósio Brasileiro de Educação em Computação (EDUCOMP)*, pages 245–255.
- de Souza, M. F. B., de Oliveira, M. G., and de Souza, M. A. V. F. (2024). Boolace: Jogo educacional para auxiliar o ensino e a aprendizagem de lógica proposicional. In *Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*, pages 235–240. SBC.
- dos Santos, G. A., Mariano, C. T., Gomes, J. A. S., Almeida, J. d. C. S., de Albuquerque Evangelista, M. E., and Santos, V. L. S. (2026). Aprendizagem colaborativa: Tecnologias educacionais e taxonomia de bloom. *Revista Ibero-Americana de Humanidades, Ciências e Educação*, 12(1):1–7.
- Ferraz, A. P. d. C. M. and Belhot, R. V. (2010). Taxonomia de bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais. *Gestão & produção*, 17:421–431.
- Garcia, L. M. L. d. S., Lara, D. F., and Gomes, R. S. (2023). As taxonomias de bloom e solo no planejamento do ensino de habilidades e competências em aprendizado de maquina. *Regae: Revista de Gestão e Avaliação Educacional*, 12(21).
- Govender, R. G. and Govender, D. W. (2023). Using robotics in the learning of computer programming: Student experiences based on experiential learning cycles. *Education Sciences*, 13(3):322.
- Johnson Jr, S. L., Bishop, J. P., and Rogers, K. D. (2025). Shifting pedagogically: Incorporating the social, cultural, and emotional dimensions of student learning to develop stem-identities in computer science. *Journal of Applied Developmental Psychology*, 97:101757.
- Kolb, D. A. (2015). *Experiential Learning: Experience as the Source of Learning and Development*. Pearson Education, New Jersey.
- Konak, A., Clark, T. K., and Nasereddin, M. (2014). Using kolb’s experiential learning cycle to improve student learning in virtual computer laboratories. *Computers & Education*, 72:11–22.
- Li, M., Wang, D., Purwanto, E., Selig, T., Zhang, Q., and Liang, H.-N. (2025). Visualcodemooc: A course platform for algorithms and data structures integrating a conversational agent for enhanced learning through dynamic visualizations. *SoftwareX*, 30:102072.
- Lima, A., Holanda, M., Borges, V. R., and Ishikawa, E. (2024). Análise do uso de componentes do moodle em uma disciplina introdutória de programação. In *Simpósio Brasileiro de Educação em Computação (EDUCOMP)*, pages 102–112. SBC.

- López-Pernas, S., Saqr, M., and Viberg, O. (2021). Putting it all together: Combining learning analytics methods and data sources to understand students' approaches to learning programming. *Sustainability*, 13(9):4825.
- Luo, X. and Adelopo, I. (2025). Exploring pedagogies, opportunities and challenges of teaching and learning programming in business school. *Journal of International Education in Business*, 18(1):26–46.
- Maia, M. C. O., Serey, D., and Figueiredo, J. (2017). Learning styles in programming education: A systematic mapping study. In *2017 IEEE Frontiers in Education Conference (FIE)*, pages 1–7. IEEE.
- Marietto, M. d. G. B., Botelho, W. T., da Motta Ferreira, J. C., and Pimentel, E. P. (2014). Teoria da aprendizagem experiencial de kolb e o ciclo de belhot guiando o uso de simulações computacionais no processo ensino aprendizagem. In *Anais do Workshop de Informática na Escola*, volume 20, pages 527–531.
- Masegosa, A. R., Cabañas, R., Maldonado, A. D., and Morales, M. (2024). Learning styles impact students' perceptions on active learning methodologies: a case study on the use of live coding and short programming exercises. *Education Sciences*, 14(3):250.
- Messer, M., Brown, N. C., Kölling, M., and Shi, M. (2024). Automated grading and feedback tools for programming education: A systematic review. *ACM Transactions on Computing Education*, 24(1):1–43.
- Pereira, M. R., Cardoso, P. C., Greggi, J. G., Uchôa, J. Q., and da Silva, R. R. (2024). Exploring a hybrid methodology: Experience report in introductory programming for computer science and information systems courses. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 72–84. SBC.
- Porto, B., Dias, D. M., and Battestin, V. (2023). Tendências de learning analytics em moodle: uma revisão sistemática. *EaD em Foco*, 13(1):e2070–e2070.
- Sanal Kumar, T. and Thandeeswaran, R. (2024). An improved adaptive personalization model for instructional video-based e-learning environments. *Journal of Computers in Education*, pages 1–47.
- Sobral, S. R. (2021). Bloom's taxonomy to improve teaching-learning in introduction to programming.
- Solis-Lastra, J., Brandão, A. A. F., and de Carvalho Albertini, B. (2024). Framework for activity development in remote laboratories ieee 1876–2019 standard based. *2024 IEEE ANDESCON*, pages 1–6.
- Tamborlin, J. R. and Piton-Gonçalves, J. Uma análise de itens de matemática do enem digital via taxonomia de bloom revisada. *Revista Brasileira de Educação em Ciências e Educação Matemática*, 8(2).
- Taylor, A. P. (2007). Programming for the internet and experiential learning: A new approach incorporating a constructed world. *International Journal of Technology and Design Education*, 17(2):217–229.
- Tzafilkou, K., Protogeros, N., and Chouliara, A. (2020). Experiential learning in web development courses: Examining students' performance, perception and acceptance. *Education and Information Technologies*, 25(6):5687–5701.

- Widiana, I. W., Triyono, S., Sudirtha, I. G., Adijaya, M. A., and Wulandari, I. G. A. A. M. (2023). Bloom's revised taxonomy-oriented learning activity to improve reading interest and creative thinking skills. *Cogent Education*, 10(2):2221482.
- Wu, Z. and Wan, S. (2025). A knowledge-driven approach to ai-based personalized test paper creation in programming education. *International Journal of Knowledge Management (IJKM)*, 21(1):1–21.