

Pensamento Computacional na Programação Introdutória e Habilidades do Século XXI: Um Mapeamento Sistemático da Literatura

Katyeudo Karlos de Sousa Oliveira - Universidade de São Paulo - karlos_oliveira@usp.br
Orcid: 0000-0002-4414-1599

Anderson da Silva Marcolino - Universidade Federal do Paraná - anderson.marcolino@ufpr.br
Orcid: 0000-0002-4014-1882

William Simão de Deus - Universidade de São Paulo - williamsimao@usp.br
Orcid: 0000-0001-7540-3742

Taciana Pontual Falcão - Universidade Federal Rural de Pernambuco - taciana.pontual@ufrpe.br
Orcid: 0000-0003-2775-4913

Ellen Francine Barbosa - Universidade de São Paulo - francine@icmc.usp.br
Orcid: 0000-0003-3275-2293

Resumo: A programação introdutória capacita estudantes a resolver problemas com soluções algorítmicas. O Pensamento Computacional (PC), definido como a habilidade de resolver problemas e projetar sistemas com base em conceitos da Ciência da Computação, desempenha um papel significativo na aprendizagem de programação e profissões do século XXI. Este artigo apresenta um Mapeamento Sistemático da Literatura que investiga o desenvolvimento das habilidades de programação introdutória e habilidades do século XXI por meio do PC. Os resultados revelaram, entre outros achados, que a linguagem de programação *Python* tem sido amplamente adotada e com forte associação com habilidades de resolução de problemas. Além disso, cursos extracurriculares com *wikis* e *Python* mostraram-se abordagens eficazes para promover habilidades introdutórias de programação. Por fim, em relação a métricas de avaliação, a resolução de problemas destacou-se como a habilidade mais valorizada.

Palavras-chave: Habilidades do Século XXI; Pensamento Computacional; Programação Introdutória

Computational Thinking in Introductory Programming and 21st Century Skills: A Systematic Literature Mapping

Abstract: Introductory programming empowers students to solve problems using algorithmic solutions. Computational Thinking (CT), defined as the ability to solve problems and design systems based on Computer Science concepts, plays a significant role in programming learning and 21st-century professions. This article presents a Systematic Literature Mapping that investigates the development of introductory programming skills and 21st-century skills through CT. The findings revealed, among other discoveries, that the Python programming language has been widely adopted and strongly associated with problem-solving skills. Additionally, extracurricular courses involving wikis and Python have proven to be effective approaches to foster introductory programming skills. Finally, concerning evaluation metrics, problem-solving emerged as the most valued skill.

Keywords: 21st Century Skills; Computational Thinking; Introductory Programming

1 Introdução

A programação introdutória é uma disciplina que visa capacitar os alunos a resolver problemas usando soluções algorítmicas (Luxton-Reilly *et al.*, 2018). O PC, conforme definido por Wing (2006), é a habilidade de resolver problemas e projetar sistemas com base em conceitos fundamentais da Ciência da Computação, sendo considerado uma competência crítica para a aprendizagem de programação (Weintrop *et al.*, 2014). Embora o PC seja frequentemente associado ao ensino básico, também é relevante para o ensino

superior, tanto como facilitador da aprendizagem de programação quanto como habilidade necessária para estudantes de cursos de computação (Silva; Falcão, 2020).

Nouri *et al.* (2020) destacam que o PC combina habilidades de programação com competências do século XXI, como pensamento crítico, resolução de problemas e autogerenciamento, fundamentais em um mundo em constante evolução tecnológica (WEF, 2020a). Essas habilidades são especialmente valorizadas em profissões emergentes, como cientistas de dados e especialistas em Inteligência Artificial (WEF 2020b). Além disso, é crucial que os estudantes do ensino superior desenvolvam tais habilidades para alcançar sucesso profissional no novo ambiente de trabalho digital (Oliveira; Souza, 2022; Oliveira *et al.*, 2022).

Nesse contexto, este artigo apresenta um Mapeamento Sistemático da Literatura (MSL), conduzido com o objetivo de investigar o desenvolvimento das habilidades de programação introdutória e habilidades do século XXI por meio do Pensamento Computacional no ensino superior.

O artigo está estruturado como se segue: a Seção 2 apresenta o protocolo adotado no MSL. Na Seção 3, são apresentados a abordagem adotada para cada questão de pesquisa e os resultados, além da análise e discussão dos resultados obtidos. Por fim, a Seção 4 apresenta as conclusões e sugestões para pesquisas futuras.

2. Protocolo

O MSL foi conduzido de acordo com as orientações de Kitchenham e Charters (2007), combinando a pesquisa automatizada utilizando *strings* de pesquisa em bibliotecas digitais com o método de *Snowballing* (Wohlin, 2014), analisando estudos que citam o estudo já selecionado anteriormente (*forward snowballing*) por meio da plataforma *Google Scholar* e conduzida avaliação por pares.

2.1. Questões de pesquisa

O principal objetivo deste MSL é identificar os estudos mais relevantes que discutem o desenvolvimento das habilidades de programação introdutória e habilidades do século XXI através do PC no ensino superior. O objetivo do MSL foi decomposto nas seguintes Questões de Pesquisa (QP): **QP1** - Quais as ferramentas utilizadas para facilitar o processo de ensino e aprendizagem das habilidades de programação introdutória no ensino superior por meio do PC?; **QP2** - Quais as abordagens utilizadas para facilitar o processo de ensino e aprendizagem das habilidades de programação introdutória no ensino superior por meio do PC?; **QP3** - Quais habilidades do século XXI as práticas de PC fomentam nos estudantes de programação do ensino superior?; **QP4** - Quais são as métricas utilizadas para avaliar o nível das habilidades de programação introdutória e as habilidades do século XXI dos estudantes de programação do ensino superior?

2.2. Estratégia da Pesquisa

A busca foi conduzida em todo o texto das publicações, não se restringindo apenas aos títulos. A *string* de busca utilizada foi: ("*Computational Thinking*") AND ("*Introductory Programming*" OR "*CS 1*" OR *CS1* OR "*Introduction to Programming*") AND (*Skill*). A organização dos termos de busca levou em consideração as peculiaridades de cada base de busca devido às diferenças de sintaxe.

Os estudos foram selecionados nas bases de dados internacionais: *ACM Digital Library*, *IEEE Xplore Digital Library*, *Science@Direct*, *Scopus* e *Springer Link*. O período de publicação considerado foi de 2012 a 2022. O ano de 2012 foi escolhido como ponto de partida devido à publicação amplamente reconhecida de Brennan e Resnick (2012). Os

autores desenvolveram um *framework* de avaliação do PC envolvendo três dimensões-chave: conceitos computacionais, práticas computacionais e perspectivas computacionais.

2.3. Critérios de Inclusão e Exclusão

Foram estabelecidos os seguintes Critérios de Inclusão (CI) e de Exclusão (CE): **Critérios de Inclusão:** **CI1** - Estudo apresenta abordagens ou ferramentas no processo de ensino e aprendizagem das habilidades de programação introdutória dos estudantes do ensino superior fomentadas pelas práticas de PC; **CI2** - Estudo apresenta pelo menos uma habilidade de programação introdutória ou uma habilidade do século XXI fomentada pelas práticas de PC; e **CI3** - Estudo apresenta pelo menos uma métrica para avaliar o nível das habilidades de programação introdutória ou habilidades do século XXI fomentadas pelas práticas de PC. **Critérios de Exclusão:** **CE1** - Estudo não é um estudo primário; **CE2** - Estudo não é escrito em língua inglesa; **CE3** - Texto completo do estudo não está acessível; e **CE4** - Estudo é duplicado (ou é versão menos detalhada/mais antiga de outro).

2.4. Condução do mapeamento sistemático da literatura

O MSL teve início em agosto de 2022. A busca inicial resultou em um total de 1050 estudos potenciais (Tabela 1).

Tabela 1. Resultados da Busca Automatizada

Base	ACM	IEEE	Science Direct	Scopus	Springer	Total
Estudos	504	328	57	55	99	1050

A aplicação dos critérios de exclusão nos 1050 estudos iniciais resultou na exclusão de 238 estudos, restando 812. Em seguida, os principais campos de identificação dos 812 estudos foram lidos visando atender aos critérios de inclusão, resultando na exclusão de 700 estudos e restando 112 para a leitura completa. Posteriormente, foi realizado um terceiro filtro de seleção com a leitura completa dos 112 estudos, resultando na exclusão de 82 estudos e selecionando 30 estudos para a etapa de *snowballing* e extração de dados. Os critérios de inclusão e exclusão foram aplicados novamente, resultando na adição de dois estudos aos 30 já selecionados. A Figura 1 apresenta o resultado final das pesquisas automáticas e *snowballing*.

Figura 1 – Iterações do mapeamento.



De cada estudo foram extraídos os seguintes dados: título, autor(es), ano de publicação, URL. Ainda, procedeu-se com a leitura completa para responder às QP propostas. O formulário de extração de dados² e os 32 estudos identificados pelo MSL³ estão disponíveis *on-line*.

3. Resultados

3.1. QP1 – Ferramentas

Considerando a QP1, este mapeamento aponta que 28 estudos utilizaram pelo menos uma ferramenta no processo de ensino e aprendizagem de habilidades de programação introdutória por meio do PC, sendo que alguns dos estudos informaram mais de uma ferramenta. No total, 36 ferramentas foram mencionadas, como apresentado na Tabela 2.

Tabela 2. Ferramentas utilizadas para facilitar o processo de ensino e aprendizagem

Ferramenta	Artigo	Ferramenta	Artigo	Ferramenta	Artigo
Linguagem de programação Python	EP05; EP08; EP09; EP11; EP24; EP25; EP27; EP29; EP30	Linguagem de programação C	EP01; EP02; EP10; EP23, EP29	Jogos digitais	EP19; EP20; EP25; EP31
Wiki	EP07; EP08; EP15; EP16	eBook	EP05; EP11	Linguagem de programação Java	EP08; EP14
Linguagem de programação MATLAB	EP07; EP08	Linguagem de programação C++	EP08; EP23	Memory Transfer Language (MTL)	EP03; EP13
Modelo preditivo	EP01; EP02	MOOC	EP06; EP25	RAPTOR	EP05; EP30
Scratch	EP26; EP31	Sistema Tutor Inteligente	EP04; EP21	YouTube	EP06; EP16
Ambiente Integrado de Desenvolvimento de Aprendizagem	EP23	App Inventor	EP26	Big Data; CORGIS (Collection of Real-time, Giant, Interesting, Situated); Google-mobwrite; NetLogo; Rhinestone	EP11
edX Online Learning; Processing.JS	EP06	Inteligência artificial; Machine Learning	EP21	Light-bot; Linguagem de programação BYOB	EP10
Linguagem de programação assembly	EP28	Linguagem de programação Processing	EP25	Medialib; Pygame	EP29
Moodle	EP32	Plataforma baseada na web	EP27	Rede neural	EP01

No EP05, foi empregada a linguagem *Python* em um ambiente de desenvolvimento visual baseado em fluxogramas e aprendizado ativo, considerando-a uma opção mais fácil para iniciantes no ensino dos conceitos básicos de programação. Em EP10, foi destacada a utilidade da linguagem de programação C no entendimento do funcionamento do computador, fundamental para programar em qualquer linguagem.

Outros estudos usaram jogos digitais como ferramentas para o aprendizado introdutório de programação. Em EP19, foi desenvolvido um *framework* de jogos digitais para praticar habilidades de PC. Em EP20, foi projetado um jogo educacional focado no desenvolvimento de habilidades de resolução de problemas na programação introdutória. No EP25, foi ressaltado o benefício dos jogos digitais no aprendizado da programação em Python e no aprimoramento das habilidades de PC. Em EP31, constatou-se que os jogos digitais podem melhorar o conhecimento de programação dos alunos, especialmente em conceitos como operadores, condicionais e *loops*.

As *wikis* foram utilizadas para facilitar a colaboração online entre grupos de alunos. Em EP15 e EP16, foram usadas wikis para acompanhar o conteúdo de programação introdutória e o envolvimento dos alunos. No EP08 foi utilizada a *wiki* para disponibilizar exercícios de criatividade computacional online. Já no EP07, foi implementado o sistema *Written Agora*, uma *wiki* que permitia a colaboração entre grupos de alunos por meio de páginas de conteúdo autoral e fóruns de discussão correspondentes.

3.2. QP2 – Abordagens

Considerando a QP2, o MSL apontou que 26 estudos utilizaram pelo menos uma abordagem para facilitar o processo de ensino e aprendizagem das habilidades de

programação introdutória por meio do PC, sendo que alguns dos estudos informaram mais de uma abordagem. No total, 35 abordagens foram mencionadas, sintetizadas na Tabela 3.

Tabela 3. Abordagens utilizadas para facilitar o processo de ensino e aprendizagem

Abordagem	Artigo	Abordagem	Artigo	Abordagem	Artigo
Curso extracurricular	EP03; EP07; EP11; EP15; EP16	Contação de histórias (<i>storytelling</i>)	EP07; EP15; EP16; EP22	Aprendizagem colaborativa	EP11; EP25; EP30
Contextos do mundo real	EP11; EP22; EP23	Criatividade computacional	EP07; EP08; EP16	Programação visual baseada em fluxograma	EP05; EP17; EP30
Teoria construtivista	EP10; EP23; EP28	Teoria da Generatividade de Epstein	EP07; EP15; EP16	Taxonomia de Bloom	EP12; EP18
Aprendizagem baseada em jogos	EP20; EP28	Aprendizagem ativa	EP27	Aprendizagem de consulta guiada	EP10
Aprendizagem interativa	EP05	<i>Blended-learning</i>	EP28	Discussão entre pares; Pedagogia spiral	EP14
Framework de Brennan e Resnick (2012); Framework TPACK; Metáforas	EP31	Gamificação	EP21	<i>Learning by teaching</i> (aprender ensinando)	EP13
Mapas de pensamento	EP25	<i>Media computation</i>	EP06	<i>Notional machine</i> ; Pedagogia hermenêutica	EP29
Pensamento computacional desconectado	EP22	Pseudocódigo; Tutoria	EP17	Serious Game	EP19
Taxonomia digital de Bloom); Taxonomia revisada de Bloom; Taxonomia SOLO	EP12	Teoria Cognitiva da Aprendizagem Multimídia); Teoria da Aprendizagem Experiencial de Kolb; Teoria da Carga Cognitiva	EP23		

A abordagem mais utilizada foi a de cursos extracurriculares, identificada em cinco estudos. Em EP03, foi desenvolvido um curso para auxiliar alunos com dificuldades em programação, enquanto em EP11 foi implementado um projeto de PC com aprendizagem colaborativa e contextos do mundo real. Em EP07, EP15 e EP16 também foram adotados os cursos extracurriculares, incorporando a Teoria da Generatividade de Epstein, Criatividade Computacional e *storytelling*, esta última mencionada em quatro estudos. Além disso, foram identificadas outras abordagens para facilitar o ensino e aprendizagem de programação introdutória com o PC no ensino superior: aprendizagem colaborativa (EP11; EP25; EP30), contextos do mundo real (EP11; EP22; EP23) e Teoria Construtivista (EP10; EP23; EP28). A aprendizagem colaborativa envolve grupos de alunos trabalhando juntos para resolver problemas, enquanto a abordagem de contextos do mundo real auxilia na aplicação de soluções de computação a problemas reais. A Teoria Construtivista enfatiza o papel ativo do aluno na construção do seu próprio conhecimento.

Os achados deste MSL mostram ainda que os cursos extracurriculares são reconhecidos por promover habilidades de programação introdutória, despertando o interesse dos alunos pela computação e preparando-os para estudos e carreiras relacionadas. Essa abordagem é comum tanto na educação básica quanto no ensino superior (EP04). *Wikis*, *Python* e *Memory Transfer Language* são as ferramentas mais utilizadas nesses cursos.

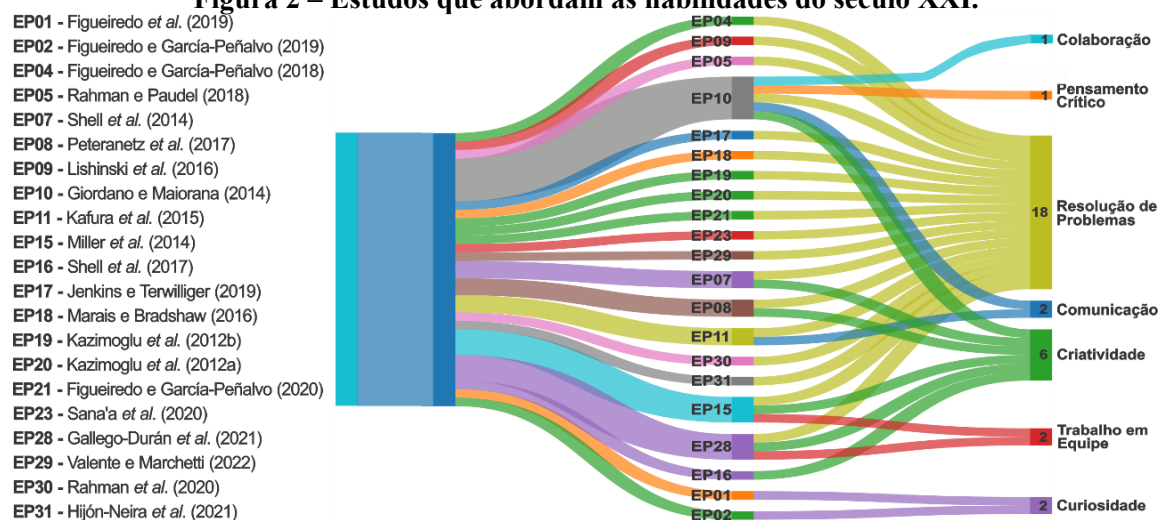
3.3. QP3 – Habilidades

De acordo com estudos realizados por organizações internacionais que mantêm fóruns de discussão sobre educação, como *The Organization for Economic Co-operation and Development* (OCDE, 2018), *The Partnership for 21st Century Learning* (P21, 2019), V. 21 N° 2, dezembro, 2023

e *The World Economic Forum* (WEF, 2015), as habilidades do século XXI são: colaboração, comunicação, criatividade, curiosidade, empatia, liderança, pensamento crítico, persistência, resolução de problemas, responsabilidade e trabalho em equipe. Portanto, na etapa de extração de dados, se algum estudo possuir alguma dessas habilidades, mesmo não sendo referida no texto como sendo uma habilidade do século XXI, a habilidade era extraída. Desse modo, em relação à QP3, dos 32 estudos analisados neste MSL, 21 apresentaram pelo menos uma habilidade do século XXI, alguns apontando mais de uma. No total, sete habilidades foram mencionadas como sendo fomentadas no processo de ensino de programação introdutória, mostradas na Figura 2.

Com 18 estudos, a habilidade de resolução de problemas foi mencionada com maior frequência. De fato, o PC é definido como uma habilidade voltada à resolução de problemas. Pesquisadores no ensino de computação se interessam pela relação entre habilidades de resolução de problemas e habilidades de programação. Um dos estudos (EP09) sugeriu que a resolução de problemas tem relação com o desempenho positivo dos alunos na programação introdutória. Outro estudo (EP10) apresentou uma abordagem educativa para o ensino de programação introdutória baseada no desenvolvimento de competências fundamentais e transversais e competências informáticas, incluindo a resolução de problemas. Porém, embora os estudos observados neste MSL foquem em sua maioria na habilidade de resolução de problemas, os alunos estão enfrentando dificuldades para adquirir essa habilidade durante os cursos introdutórios de programação (Medeiros *et al.*, 2021).

Figura 2 – Estudos que abordam as habilidades do século XXI.



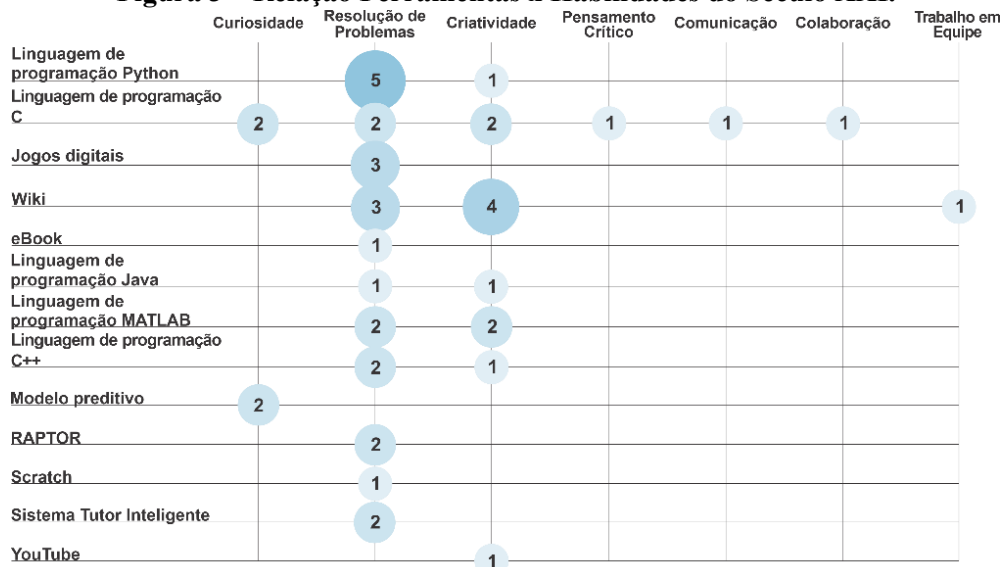
Já a criatividade foi mencionada em seis estudos. O PC apoia a criatividade dos indivíduos e a inclusão de atividades que englobam o aspecto criativo pode aprimorar o aprendizado em disciplinas introdutórias de programação. Os estudos conduzidos por EP16 e EP07 revelam que a inclusão de atividades criativas melhora o aprendizado em disciplinas introdutórias de programação. Esses achados sugerem que o desenvolvimento das habilidades de programação está relacionado ao estímulo das competências criativas.

Diferentes autores indicam uma grande variedade de habilidades relacionadas ao aprendizado de programação, e há um interesse crescente em entender detalhadamente a natureza do aprendizado de programação em conjunto com as habilidades do século XXI. Uma análise das relações entre as ferramentas e as habilidades do século XXI (Figura 3) mostra que a linguagem de programação Python tem mais referências em comum com a habilidade de resolução de problemas (a mais citada nos estudos deste MSL).

Jogos digitais e *wikis* foram outras ferramentas mencionadas, sendo os jogos digitais associados à resolução de problemas e as *wikis* principalmente à criatividade. O *Scratch*, por sua vez, é uma ferramenta popular para iniciantes, mas estudos mostram que

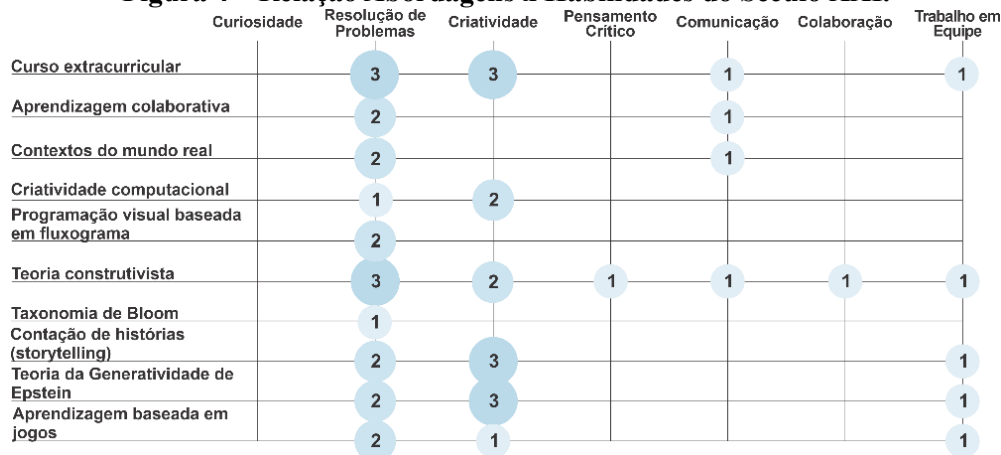
pode não ser atrativo para estudantes universitários com experiência anterior em programação. A escolha da ferramenta mais adequada e sua relação com as habilidades do século XXI requerem uma investigação mais aprofundada.

Figura 3 – Relação Ferramentas x Habilidades do Século XXI.



A habilidade de resolução de problemas está relacionada a todas as abordagens mencionadas, enquanto a teoria construtivista destaca a conexão com as habilidades do século XXI, especialmente a resolução de problemas e a criatividade (Figura 4). Apesar dos cursos extracurriculares serem a abordagem mais citada para o ensino de habilidades de programação introdutória por meio do PC, mais pesquisas são necessárias para determinar a melhor abordagem e combinações adequadas para as necessidades e contextos do ensino de programação introdutória. A amostra de estudos atual é relativamente pequena para fazer afirmações conclusivas.

Figura 4 – Relação Abordagens x Habilidades do Século XXI.



3.4. QP4 – Métricas

A QP4 aborda as métricas utilizadas para avaliar o nível das habilidades de programação introdutória e as habilidades do século XXI dos estudantes de programação do ensino superior. Dos 32 estudos selecionados, quatro não indicaram nenhuma métrica. Os demais trabalhos apresentaram, ao todo, 17 métricas, sumarizadas na Tabela 4.

Três estudos primários (EP01; EP02 e EP03) adotaram a métrica de avaliação de desempenho para avaliar as atividades de programação dos alunos e identificar suas dificuldades e limitações precocemente, a fim de prevenir reprovações ou desistências. Esses estudos desenvolveram um modelo preditivo baseado em redes neurais, utilizando o

perfil dos alunos e coletando informações ao longo das aulas. Além disso, outros estudos (EP11; EP14; EP15; EP16; EP17; EP22 e EP23) também utilizaram o desempenho dos alunos como métrica de avaliação, empregando diferentes metodologias, como projetos e testes (EP11; EP15; EP16; EP17; EP23), avaliação do PC relacionada às motivações dos alunos (EP22; EP23), e método pedagógico *Spiral* (EP14). Outras abordagens incluem exercícios de criatividade computacional (EP07; EP08; EP16) e a integração de feedback imediato ao processo de aprendizagem (EP17), por meio de atividades práticas e colaborativas.

Tabela 4. Métricas de avaliação das habilidades de programação e habilidades do século XXI

Métrica	Artigo	Métrica	Artigo	Métrica	Artigo
Desempenho dos alunos;	EP01; EP02; EP03; EP09; EP11; EP14; EP15; EP16; EP17; EP22; EP23	Pós-teste	EP05; EP06; EP10; EP19; EP20; EP22; EP25; EP28; EP29; EP30; EP31	Pré-teste	(EP06; EP10; EP22; EP25; EP28; EP29; EP30; EP31)
Student Perceptions of Classroom Knowledge Building (SPOCK)	EP07; EP08	Avaliação por pares	EP06	(Bebras); (Talent Search Computing Olympiad)	EP32
Entrevistas presenciais; Fórum de discussão da comunidade de prática; Questionário online	EP12	Feedback imediato	EP17	Constructo de Hoegh e Moskal, (2009); Constructo de Yadav <i>et al.</i> , (2014)	EP26
Pesquisa indutiva	EP29	Positive and Negative Affect Scale (PANAS)	EP08	Teoria clássica dos testes	EP24
Testes de Pensamento Computacional	EP18				

Diversos autores (EP23; EP06; EP10; EP22; EP25; EP28; EP29; EP30; EP31; EP07 e EP08) utilizam diferentes métricas e métodos, tais como pré e pós-testes, avaliação por pares, pesquisa indutiva, exercícios de criatividade computacional, *Student Perceptions of Classroom Knowledge Building* (SPOCK) e *Positive and Negative Affect Scale* (PANAS).

Alguns estudos utilizam questionários online, fóruns, entrevistas, testes de habilidades e questões de olimpíadas de programação como métodos de avaliação. No entanto, é importante que os pesquisadores disponibilizem detalhes adicionais e compartilhem pacotes experimentais e de avaliação para permitir um melhor aproveitamento dessas métricas. Além disso, a falta de identificação dos fundamentos teóricos e formalismos nas métricas mais utilizadas, como desempenho dos alunos, pré e pós-teste, levanta questões sobre a confiabilidade de alguns estudos, que poderiam ser evitadas com a disponibilização ou indicação do processo de desenvolvimento desses instrumentos. A Figura 5 relaciona essas métricas com as habilidades do Século XXI, sendo a resolução de problemas a mais avaliada, por meio das sete métricas identificadas. A métrica predominante para a avaliação da resolução de problemas foi o Pós-teste. No entanto, é importante que os questionários sejam rigorosos para garantir a avaliação adequada da resolução do problema.

A curiosidade foi avaliada usando métricas como desempenho dos alunos, pós-teste e pré-teste. Para avaliar a curiosidade de forma precisa, é importante definir critérios claros e considerar a sensibilidade do avaliador. No caso da habilidade de criatividade, utilizaram-se métricas como desempenho dos alunos, SPOCK (EP07; EP08) e PANAS (EP08). O SPOCK avalia a construção do conhecimento e a aprendizagem colaborativa, enquanto o

PANAS mede estados afetivos positivos e negativos, proporcionando uma avaliação abrangente que considera o estado emocional dos estudantes. A habilidade de comunicação foi avaliada de maneira semelhante à curiosidade, utilizando métricas como desempenho dos alunos, pós-teste e pré-teste, além do *feedback* dos colegas e dos próprios alunos por meio de questionários. Para as habilidades de pensamento crítico, colaboração e trabalho em equipe, aplicaram-se os mesmos métodos de avaliação. É essencial compreender as particularidades de cada habilidade e garantir que as métricas englobem todas as nuances envolvidas para uma avaliação completa e precisa.

Figura 5 – Relação Métricas x Habilidades do Século XXI.



4. Conclusões e Trabalhos Futuros

Neste artigo foram apresentados e discutidos os resultados de um MSL conduzido com o objetivo de investigar o desenvolvimento das habilidades de programação introdutória e habilidades do século XXI por meio do PC no ensino superior. Com base nos resultados, pode-se concluir que o PC desempenha um papel crucial no desenvolvimento de habilidades introdutórias de programação e habilidades do século XXI. Ao analisar as respostas às quatro QPs, foram identificadas tendências e relações importantes.

A QP1 revelou que *Python* é amplamente adotada e está fortemente ligada à resolução de problemas. Abordagens como cursos extracurriculares com *wikis* e *Python*, e o uso de jogos digitais, foram identificadas como promissoras para promover habilidades introdutórias de programação e aprimorar o conhecimento dos alunos nessa área. Na QP2, os cursos extracurriculares surgiram como a abordagem mais comum para facilitar o ensino de programação introdutória com PC. Além disso, abordagens como aprendizagem colaborativa e contextos do mundo real mostraram-se eficazes na promoção de habilidades de programação. Na QP3, verificou-se que as habilidades mais enfatizadas no ensino de programação introdutória são a resolução de problemas e a criatividade, as quais estão intrinsecamente ligadas ao PC e consideradas fundamentais para a programação. Por fim, na QP4, foram identificadas diversas métricas de avaliação utilizadas para mensurar as habilidades de programação introdutória e as habilidades do século XXI dos estudantes. Entre as métricas mais comuns estão o desempenho dos alunos, pré e pós-testes.

Este MSL tem limitações devido à subjetividade. Para mitigá-las, foram adotadas ações como seleção cuidadosa dos estudos seguindo um protocolo bem definido com base em diretrizes rígidas (Kitchenham; Charters 2007), busca em diferentes bibliotecas digitais e técnicas complementares de seleção (*snowballing*), bem como avaliação por pares. As habilidades do século XXI foram definidas com base em publicações de fóruns internacionais (WEF, 2015; OECD, 2018; P21, 2019).

Destarte, foram identificadas diversas lacunas nas pesquisas obtidas, como escolha da abordagem, avaliação das habilidades dos alunos e exploração restrita à algumas habilidades do século XXI na programação introdutória. Adicionalmente, os resultados

discutidos variaram conforme o contexto educacional e as características dos alunos. Portanto, pesquisas futuras devem investigar estratégias pedagógicas, métricas confiáveis e formação de professores para promover as habilidades do século XXI na programação.

Notas

¹CS1 refere-se a cursos que se concentram em capacitar os estudantes com conceitos de programação introdutória usando linguagens de programação (Luxton-Reilly *et al.*, 2018)

²Formulário de extração de dados: <https://x.gd/i7zws>

³Estudos identificados pelo MSL: <https://x.gd/9BaQ5>

Agradecimentos

Os autores agradecem ao apoio financeiro da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código Financeiro 001, do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP - Processo 2019/26871-4), e da Especialização em Computação Aplicada à Educação (CAE - ICMC/USP).

Referências

- ANISTYASARI, Y.; EKOARIADI, E.; BUDITJAHJANTO, I. P. A.; HIDAYATI, S. C. Exploring the psychometric properties of computational thinking assessment in introductory programming. In: **2021 International e-Engineering Education Services Conference (e-Engineering)**. IEEE, 2021. p. 88-93.
- BRENNAN, K.; RESNICK, M. New frameworks for studying and assessing the development of computational thinking. In: **Proceedings of the 2012 annual meeting of the American educational research association**, Vancouver, Canada. 2012. p. 25.
- EBERT, M. Increase active learning in programming courses. In: **2017 IEEE Global Engineering Education Conference (EDUCON)**. IEEE, 2017. p. 848-851.
- FALKNER, K.; FALKNER, N.; SZABO, C.; VIVIAN, R. (2016). Applying validated pedagogy to MOOCs: An introductory programming course with media computation. In: **Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education**. 2016. p. 326-331.
- FIGUEIREDO, J.; GARCÍA-PEÑALVO, F. J. Improving computational thinking using follow and give instructions. In: **Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality**. 2017. p. 1-7.
- FIGUEIREDO, J.; GARCÍA-PEÑALVO, F. J. Building skills in introductory programming. In: **Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality**. 2018. p. 46-50.
- FIGUEIREDO, J.; GARCÍA-PEÑALVO, F. J. Teaching and learning strategies of programming for university courses. In: **Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality**. 2019. p. 1020-1027.
- FIGUEIREDO, J.; GARCÍA-PEÑALVO, F. J. Intelligent tutoring systems approach to introductory programming courses. In: **Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality**. 2020. p. 34-39.
- FIGUEIREDO, J.; GOMES, N.; GARCÍA-PEÑALVO, F. J. Ne-course for learning programming. In: **Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality**. 2016. p. 549-553.
- FIGUEIREDO, J.; LOPES, N.; GARCÍA-PEÑALVO, F. J. Predicting student failure in an introductory programming course with multiple back-propagation. In: **Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality**. 2019. p. 44-49.

GALLEGO-DURÁN, F. J.; SATORRE-CUERDA, R.; COMPAÑ-ROSIQUE, P.; VILLAGRÁ-ARNEDO, C. J.; MOLINA-CARMONA, R.; LLORENS-LARGO, F. A low-level approach to improve programming learning. **Universal Access in the Information Society**, v. 20, p. 479-493, 2021.

GIORDANO, D.; MAIORANA, F. Use of cutting edge educational tools for an initial programming course. In: **2014 IEEE Global Engineering Education Conference (EDUCON)**. IEEE, 2014. p. 556-563.

HIJÓN-NEIRA, R.; CONNOLLY, C.; PALACIOS-ALONSO, D.; BORRÁS-GENÉ, O. A guided Scratch visual execution environment to introduce programming concepts to CS1 students. **Information**, v. 12, n. 9, p. 378, 2021.

JENKINS, J. T.; TERWILLIGER, M. G. Examining strategies to improve student success in CS1. **Journal of Computing Sciences in Colleges**, v. 35, n. 4, p. 124-132, 2019.

KAFURA, D.; BART, A. C.; CHOWDHURY, B. Design and preliminary results from a computational thinking course. In: **Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education**. 2015. p. 63-68.

KAZIMOGLU, C.; KIERNAN, M.; BACON, L.; MACKINNON, L. Learning programming at the computational thinking level via digital game-play. **Procedia Computer Science**, v. 9, p. 522-531, 2012a.

KAZIMOGLU, C.; KIERNAN, M.; BACON, L.; MACKINNON, L. A serious game for developing computational thinking and learning introductory computer programming. **Procedia-Social and Behavioral Sciences**, v. 47, p. 1991-1999, 2012b.

KITCHENHAM, B.; CHARTERS, S. Guidelines for performing Systematic Literature Reviews in Software Engineering, Technical Report. **EBSE 2007-001**. Keele University and Durham University Joint Report. 2007.

LIONELLE, A.; GHOSH, S.; SAY, B.; BEVERIDGE, J. R. Increase Performance in CS 2 via a Spiral Redesign of CS 1. In: **Proceedings of the 53rd ACM Technical Symposium on Computer Science Education-Volume 1**. 2022. p. 502-508.

LISHINSKI, A.; YADAV, A.; ENBODY, R.; GOOD, J. The influence of problem solving abilities on students' performance on different assessment tasks in CS1. In: **Proceedings of the 47th ACM technical symposium on computing science education**. 2016. p. 329-334.

LUXTON-REILLY, A.; ALBLUWI, I.; BECKER, B. A.; GIANNAKOS, M.; KUMAR, A. N.; OTT, L.; SZABO, C. Introductory programming: a systematic literature review. In: **Proceedings companion of the 23rd annual ACM conference on innovation and technology in computer science education**. 2018. p. 55-106.

MARAIS, C.; BRADSHAW, K. Towards a technical skills curriculum to supplement traditional computer science teaching. In: **Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education**. 2016. p. 338-343.

MEDEIROS, R. P.; FALCÃO, T. P.; RAMALHO, G. L. Comparação entre o panorama internacional e nacional sobre o Ensino e a Aprendizagem de Introdução à Programação no Ensino Superior. In: **Anais do XXIX Workshop sobre Educação em Computação**. SBC, 2021. p. 478-487.

MILLER, L. D.; SOH, L. K.; CHIRIACESCU, V.; INGRAHAM, E.; SHELL, D. F.; HAZLEY, M. P. Integrating computational and creative thinking to improve learning and performance in CS1. In: **Proceedings of the 45th ACM technical symposium on Computer science education**. 2014. p. 475-480.

NOURI, J.; ZHANG, L.; MANNILA, L.; NORÉN, E. Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. **Education Inquiry**, v. 11, n. 1, p. 1-17, 2020.

OECD. Organisation for Economic Co-operation and Development. The future of education and skills: Education 2030. OECD Education 2030. **OECD Publishing**, Paris. 2018.

- OLIVEIRA, E. C.; CORREIA, R. C. M.; BITTENCOURT, R. A. Evaluating a computational thinking and computing attitudes instrument for educational purposes. In: **2021 IEEE Global Engineering Education Conference (EDUCON)**. IEEE, 2021. p. 748-754.
- OLIVEIRA, K. K. S.; FALCÃO, T. P.; BARBOSA, E. F. Um framework para a aprendizagem de habilidades de programação e habilidades do século XXI. In: **Anais Estendidos do XI Congresso Brasileiro de Informática na Educação**. SBC, 2022. p. 138-143.
- OLIVEIRA, K. K. S.; SOUZA, R. A. C. Digital transformation towards education 4.0. **Informatics in Education**, v. 21, n. 2, p. 283-309, 2022.
- P21. Partnership for 21st Century Learning. **Partnership for 21st Century Learning: Framework for 21st Century Learning Definitions**. Battelle for Kids. 2019.
- PARHAM-MOCELLO, J.; ERWIG, M.; DOMINGUEZ, E. To code or not to code? Programming in introductory CS courses. In: **2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)**. IEEE, 2019. p. 187-191.
- PETERANETZ, M. S.; FLANIGAN, A. E.; SHELL, D. F.; SOH, L. K. Computational creativity exercises: An avenue for promoting learning in computer science. **IEEE Transactions on Education**, v. 60, n. 4, p. 305-313, 2017.
- RAHMAN, M. M.; PAUDEL, R. Visual programming and interactive learning based dynamic instructional approaches to teach an introductory programming course. In: **2018 IEEE Frontiers in Education Conference (FIE)**. IEEE, 2018. p. 1-6.
- RAHMAN, M. M.; SHARKER, M. H.; PAUDEL, R. An Effective Approach to Teach an Introductory Computer Science Course with Computational Thinking and Flow-Chart Based Visual Programming. In: **2020 IEEE Frontiers in Education Conference (FIE)**. IEEE, 2020. p. 1-7.
- ROJAS-LÓPEZ, A.; GARCÍA-PENÁLVO, F. J. Learning scenarios for the subject methodology of programming from evaluating the computational thinking of new students. **IEEE Revista Iberoamericana de Tecnologías del Aprendizaje**, v. 13, n. 1, p. 30-36, 2018.
- SELBY, C. C. Relationships: computational thinking, pedagogy of programming, and Bloom's Taxonomy. In: **Proceedings of the workshop in primary and secondary computing education**. 2015. p. 80-87.
- SANA'A, M. A.; DOUSAY, T. A.; JEFFERY, C. L. Integrated learning development environment for learning and teaching C/C++ language to novice programmers. In: **2020 IEEE Frontiers in Education Conference (FIE)**. IEEE, 2020. p. 1-5.
- SHELL, D. F.; HAZLEY, M. P.; SOH, L. K.; MILLER, L. D.; CHIRIACESCU, V.; INGRAHAM, E. Improving learning of computational thinking using computational creativity exercises in a college CSI computer science course for engineers. In: **2014 IEEE frontiers in education conference (FIE) proceedings**. IEEE, 2014. p. 1-7.
- SHELL, D. F.; SOH, L. K.; FLANIGAN, A. E.; PETERANETZ, M. S.; INGRAHAM, E. Improving students' learning and achievement in CS classrooms through computational creativity exercises that integrate computational and creative thinking. In: **Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education**. 2017. p. 543-548.
- SILVA, E. O.; FALCÃO, T. P. O pensamento computacional no ensino superior e seu impacto na aprendizagem de programação. In: **Anais do XXVIII Workshop sobre Educação em Computação**. SBC, 2020. p. 171-175.
- VALENTE, A.; MARCHETTI, E. Simplifying Programming for Non-technical Students: A Hermeneutic Approach. **KI-Künstliche Intelligenz**, p. 1-17, 2022.
- WEF. World Economic Forum. **New Vision for Education: Unlocking the Potential of Technology**. Switzerland, 2015.
- WEF. World Economic Forum. **Schools of the Future. Defining New Models of Education for the Fourth Industrial Revolution**. In **Platform for Shaping the Future of the New Economy and Society**. Cologny/Geneva, Switzerland, 2020a.

WEF. World Economic Forum. **The Future of Jobs Report 2020**. Cologny/Geneva, Switzerland, 2020b.

WEINTROP, D.; BEHESHTI, E.; HORN, M. S.; ORTON, K.; TROUILLE, L.; JONA, K.; WILENSKY, U. Interactive assessment tools for computational thinking in high school STEM classrooms. In: **Intelligent Technologies for Interactive Entertainment: 6th International Conference, INTETAIN 2014, Chicago, IL, USA, July 9-11, 2014. Proceedings 6**. Springer International Publishing, 2014. p. 22-25.

WING, J. M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33-35, 2006.

WOHLIN, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: **Proceedings of the 18th international conference on evaluation and assessment in software engineering**. 2014. p. 1-10.

XU, Z.; SHEN, X.; LIN, S.; ZHANG, F. Using Visualization to Teach an Introductory Programming Course with Python. In: **2021 11th International Conference on Information Technology in Medicine and Education (ITME)**. IEEE, 2021. p. 514-518.