

## **Autoavaliação na aprendizagem de programação introdutória: uma revisão sistemática da literatura**

### **Self-assessment in introductory programming learning: a systematic literature review**

DOI:10.34117/bjdv8n5-001

Recebimento dos originais: 21/03/2022

Aceitação para publicação: 29/04/2022

#### **Bárbara Gabriela Sousa da Silva**

Graduanda em Tecnologia da Informação

Instituição: Bolsista da Universidade Federal Rural do Semi-Árido (UFERSA)

Centro Multidisciplinar de Pau dos Ferros (CMPF)

Endereço: BR 226, Km 405-Bairro São Geraldo, Pau dos Ferros, RN, CEP: 59900-000

E-mail: barbara.silva12970@alunos.ufersa.edu.br

#### **Laysa Mabel de Oliveira Fontes**

Doutora em Engenharia Elétrica e de Computação

Instituição: Docente da Universidade Federal Rural do Semi-Árido (UFERSA)

Endereço: BR 226, Km 405-Bairro São Geraldo, Pau dos Ferros, RN, CEP: 59900-000

E-mail: mabel.fontes@ufersa.edu.br

#### **Brígido Conrado de Brito Freitas**

Graduado em Tecnologia da Informação

Instituição: Bolsista da Universidade Federal Rural do Semi-Árido (UFERSA)

Endereço: BR 226, Km 405-Bairro São Geraldo, Pau dos Ferros, RN, CEP: 59900-000

E-mail: brigido.freitas@alunos.ufersa.edu.br

#### **RESUMO**

O processo de aprendizagem de programação em cursos introdutórios é desafiador para alunos, professores e instituições de ensino, já que, devido a diversas dificuldades, os alunos tendem a evadir dos cursos por não se acharem capazes de aprender os conceitos de programação ou sentirem dificuldades durante seu aprendizado que não são supridas pelos professores. Diante desse contexto, este artigo busca, por meio de uma revisão sistemática da literatura, fazer um levantamento sobre o estado atual e as principais características da autoavaliação na aprendizagem de programação introdutória, para que os resultados levantados possam ser utilizados em contextos reais de sala de aula, com a finalidade de prover uma melhoria no processo de ensino e aprendizagem. O protocolo seguido, permitiu a busca de artigos em bases conhecidas na área de informática para educação, onde foram aplicados os critérios de inclusão e exclusão para posterior análise de qualidade e seleção final para leitura na íntegra desses estudos, para que assim, pudesse ser feita a extração dos dados que responderam às questões de pesquisa da obra. Os resultados apontam que nos últimos 5 anos: (i) a principal abordagem de autoavaliação utilizada foi o uso de feedback personalizado; (ii) a principal técnica utilizada foi o feedback; (iii) as principais limitações reportadas pelos autores foram limitações relacionadas à abordagem utilizada; (iv) dentre os benefícios, o que obteve mais relevância foi a melhoria na compreensão do conteúdo; (v) a abordagem de medição mais

adotada foi a análise de performance; e (vi) as tecnologias que mais auxiliam no processo de autoavaliação, foram as ferramentas.

**Palavras-chave:** aprendizagem, autoavaliação, programação introdutória.

## ABSTRACT

The process of learning programming in introductory courses is challenging for students, teachers and educational institutions, since, due to various difficulties, students tend to evade the courses because they do not think they are able to learn the programming concepts or experience difficulties during the course. their learning that is not supplied by teachers. Given this context, this article seeks, through a systematic review of the literature, to survey the current state and main characteristics of self-assessment in introductory programming learning, so that the results obtained can be used in real classroom contexts, in order to provide an improvement in the teaching and learning process. The protocol followed allowed the search for articles in known databases in the area of information technology for education, where the inclusion and exclusion criteria were applied for later quality analysis and final selection for reading in full, of these studies so that it could be done the extraction of data that answered the research questions of the work. The results show that in the last 5 years: (i) the main self-assessment approach used was the use of personalized feedback; (ii) the main technique used was feedback; (iii) the main limitations reported by the authors were limitations related to the approach used; (iv) among the benefits the one that obtained the most relevance was the improvement in the understanding of the content; (v) the most adopted measurement approach was the performance analysis; and (vi) the technologies that most help in the self-assessment process were the tools.

**Keywords:** learning, self-assessment, programming.

## 1 INTRODUÇÃO

Em cursos de tecnologia que envolvem programação, como Tecnologia da Informação, Ciência da Computação e afins, têm-se um conjunto de disciplinas pelas quais os estudantes passam para alcançar sua formação. As disciplinas de introdução à programação podem ser consideradas a base para o desenvolvimento lógico e de conceitos importantes que serão usados em disciplinas mais avançadas da graduação (FRANÇA; TEDESCO, 2014).

Não se pode negar que há inúmeros desafios na aprendizagem de programação que resultam em altos índices de reprovação, desistência e, conseqüentemente, evasão dos cursos de tecnologia (PASCOAL; DE BRITO; DO REGO, 2015). As dificuldades citadas passam pela metodologia de ensino, que colabora para que o aluno, muitas vezes, fique apenas parado observando, porém, sem entender muito do que está sendo dito, por não estar envolvido no processo, acaba desmotivado e, por fim, não aprende. Por esse motivo, em ambientes de aprendizagem, é importante observar o comportamento do

aluno, evitando que o aluno acabe se frustrando ao acreditar que não é capaz de aprender (REZENDE; MESQUITA, 2017).

Por isso, faz-se necessário o uso de metodologias de ensino que apoiem a resolução de problemas como os citados, que sejam eficazes para a melhoria da experiência dentro da sala de aula e, principalmente, nas avaliações, que são o principal meio de qualificar o conhecimento adquirido pelos estudantes (FRANCISCO; MORAES, 2013).

Existem várias formas de estabelecer um processo de regulação dentro de um sistema educacional que, em sua maioria, se trata do processo de avaliação, onde o significado mais usual de avaliação é dar notas, atribuir uma classificação, integrada numa escala, equivalente a uma medida (PACHECO, 1996). Dentre os vários métodos regulativos, encontra-se a autoavaliação. O seu conceito é o de aplicar uma avaliação em si mesmo com a finalidade de buscar melhorias nos aspectos avaliados. A autoavaliação auxilia o discente a adquirir a capacidade de analisar suas próprias responsabilidades, comportamentos, pontos fortes e fracos e, também, suas necessidades para atingir objetivos (VIEIRA, 2013).

Existem evidências que apontam que práticas autoavaliativas podem ser utilizadas para melhorar a aprendizagem dos discentes (NGAI et al., 2010). Também se nota que a autoavaliação possibilita que o discente seja mais autônomo nas decisões sobre o que priorizar para seus estudos e desenvolva uma maior consciência acerca do seu desempenho em seus trabalhos (VIEIRA, 2017). Através dela, ainda, os alunos podem trazer situações reais, mais bem definidas, acerca de conteúdos ministrados pelos professores, fazendo com que os docentes sejam mais eficazes na hora de esclarecer dúvidas com os alunos sobre tópicos das aulas (PAIVA, 2017).

Todavia, ao avaliar a literatura disposta sobre a autoavaliação na aprendizagem de programação introdutória, não é possível localizar muitos estudos atualizados que façam um panorama completo e detalhado sobre o assunto, o que cria a oportunidade de uma Revisão Sistemática da Literatura (RSL) sobre o assunto, uma vez que se trata de um método bem definido, suficientemente detalhado e com alto rigor de condução. Assim, este trabalho objetivou conduzir uma RSL para fazer um levantamento sobre as principais características da autoavaliação na aprendizagem de programação introdutória.

O resto deste trabalho está organizado da seguinte forma: a Seção 2 apresenta a fundamentação teórica utilizada para o desenvolvimento deste trabalho; a Seção 3 apresenta os materiais e métodos adotados na pesquisa; a Seção 4 apresenta os resultados

e discussões; a Seção 5 discute os trabalhos correlatos; e, por fim, a Seção 6 apresenta as considerações finais e os trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Esta seção está organizada da seguinte forma: a Subseção 2.1 apresenta uma contextualização sobre revisão sistemática da literatura e a Subseção 2.2 apresenta uma visão geral sobre autoavaliação.

### 2.1 REVISÃO SISTEMÁTICA DA LITERATURA

Uma RSL consiste em uma pesquisa na qual reúnem-se, por meio de critérios de inclusão e exclusão, os estudos primários com objetivo de estudar seus resultados e verificar se esses resultados estão de forma clara, precisa e condizente com o assunto. Desse modo, uma RSL tem por finalidade sintetizar os trabalhos, levando em consideração sua relevância e conteúdo, além de reduzir o tempo de pesquisas posteriores, pois com o novo estudo secundário oriundo do primário, existirá uma atualização nos estudos mais relevantes (FALBO, 2018).

As razões para se realizar uma RSL são muitas, dentre elas: (i) sintetizar evidências que já existem sobre algum assunto de pesquisa específico; (ii) verificar se os estudos publicados estão condizentes com o tema; (iii) preencher algumas lacunas que ainda não foram cobertas sobre um determinado assunto; e (iv) gerar um segundo estudo derivado do primário a fim de melhorar pesquisas futuras, pois estarão atualizadas. Devido alguns estudos possuírem pouco valor científico e embasamento, faz-se necessário uma RSL com o objetivo final de manter-se atualizado com os melhores estudos e trabalhos (KITCHENHAM; CHARTERS, 2007).

Desse modo, uma RSL busca obter um grande número de trabalhos, denominados de trabalhos primários, dentro de um determinado escopo de pesquisa, com intuito de construir um trabalho, denominado de trabalho secundário, obedecendo critérios rígidos para o desenvolvimento de um estudo que poderá ser utilizado para fins de pesquisa. Assim, por seguir um método, ser bem planejada e justificada, uma RSL se diferencia de revisões tradicionais e, por isso, acaba por ser de extrema relevância para uma boa pesquisa (BRIZOLA; FANTIN, 2016).

## 2.2 AUTOVALIAÇÃO

A avaliação remete automaticamente ao ambiente escolar. A avaliação sempre foi vista como algo repressivo, principalmente, pelos alunos, que são avaliados mediante seu desempenho e classificados como aprovados ou reprovados ao final de um ano letivo. O método de avaliação é uma prática tradicional que, geralmente, vem empregada por meio de provas onde o aluno demonstra seu conhecimento acerca de um determinado assunto ministrado pelo professor. Essa abordagem, um pouco ultrapassada, faz com que o desenvolvimento de novas metodologias seja dificultado, atrasando o desenvolvimento e a criatividade dos discentes em um ambiente escolar (SILVA, 2016).

Essa abordagem de avaliação, resulta em altos índices de notas baixas e isso gera questionamentos acerca do principal papel da avaliação no âmbito educacional, uma vez que a intenção é promover o conhecimento ao discente de forma clara para melhorar o aprendizado do conteúdo. O método, comumente utilizado, acaba por desmotivar os alunos, desvalorizar os docentes e enfraquecer as instituições de ensino (SILVA, 2016).

Em contraste, entende-se por autoavaliação o processo de examinar os resultados obtidos em alguma atividade, fazer uma classificação desses resultados, categorizar os pontos examinados, dividindo os conteúdos da atividade em tópicos, por exemplo, e, em seguida, regular o aprendizado com base nos critérios utilizados na medição e, por fim, qualificar o conhecimento adquirido e os pontos fortes e fracos de modo a entender melhor sobre si mesmo e a maneira que se aprende os conteúdos (LOPES, 2018).

Portanto, a autoavaliação consiste no processo de avaliação das produções, ações, conduta, capacidades, gostos, desempenho, competências e habilidades do próprio aluno sobre si mesmo, com um olhar crítico, para que possa identificar o porquê de suas falhas. O objetivo é o desenvolvimento pessoal e cognitivo, bem como a tomada de estratégias para a melhoria contínua. A responsabilidade do professor, enquanto mediador do conhecimento, é incentivar os alunos ao processo autoavaliativo e utilizar os resultados para regular o processo pedagógico, sem penalizar eles (MARXREITER; BRESOLIN; FREIRE, 2021).

Com a finalidade de regulação da forma avaliativa, Vieira (2013) lista os seguintes instrumentos: (i) relatório escrito; (ii) portfólio; (iii) teste em duas fases; (iv) observação; (v) interação professor-aluno; (vi) ferramentas web (Vialogues, Wordle, Moodle, Prezi, WiziQ, GoogleDocs e Youtube). Vale a pena mencionar também os chamados *Online Judge* que podem melhorar a forma do professor trabalhar com o material didático,

criando bancos de questões para os alunos resolverem exercícios e obterem *feedback* imediato para frisarem melhor o conteúdo (FRANCISCO et. al, 2018).

Dentre os benefícios que a autoavaliação traz para os alunos, pode-se mencionar: (i) influência no futuro profissional, considerando que ele poderá refletir sobre os resultados de suas próprias ações; (ii) indicação do nível de compreensão do aluno acerca de uma determinada atividade ou conteúdo, já que, ao fazer a reflexão sobre si mesmo, o aluno é capaz de determinar se absorveu corretamente o que lhe foi passado, além de contribuir para o crescimento pessoal, pois ao se autoavaliar, os alunos poderão elaborar planos de ação para lidar com suas dificuldades, até mesmo em um contexto geral (CAPELLATO et al., 2020).

### 3 PROTOCOLO DA RSL

Nesta seção será apresentado o protocolo para a produção da RSL, seguindo os critérios propostos por Kitchenham et al. (2007).

#### 3.1 QUESTÕES DE PESQUISA

As questões de pesquisa definidas para a RSL proposta, que estão definidas no Quadro 1, foram levantadas no período de agosto de 2021 e têm como base sanar questões no contexto da autoavaliação na aprendizagem de programação introdutória.

Quadro 1: Questões de pesquisa.

<b>QP1</b>	Quais abordagens podem ser utilizadas para a autoavaliação na aprendizagem de programação introdutória?
<b>QP2</b>	Qual a técnica de autoavaliação é mais utilizada na aprendizagem de programação introdutória?
<b>QP3</b>	Quais as principais limitações reportadas por autores na aprendizagem de programação introdutória?
<b>QP4</b>	De que forma os estudos medem a autoavaliação em um cenário real de aprendizagem de programação introdutória?
<b>QP5</b>	Quais os benefícios que a autoavaliação traz para o aluno?
<b>QP6</b>	Como a tecnologia tem ajudado a autoavaliação no processo de aprendizagem de programação introdutória?

Fonte: Autoria Própria.

A busca pelas respostas das questões de pesquisa será o foco principal do processo de condução desta RSL.

### 3.2 STRINGS DE BUSCA

A pesquisa por estudos primários foi realizada através de *strings* de busca, definidas por meio de uma sondagem em bases de busca conhecidas. Para compor tais *strings*, utilizou-se operadores booleanos, *AND* e *OR*, para relacionar os termos e construir o sentido da busca da pesquisa, que foi realizada no período de setembro de 2021. O Quadro 2 relaciona as bases com as *strings* de busca utilizadas.

Quadro 2: *Strings* de busca.

Base	<i>Strings</i>
IEEE Xplore	(self-assessment) AND (programming) AND (beginner OR novice OR introductory course OR CS1 OR apprentice OR student OR learner OR algorithm)
ACM Digital Library	(self-assessment) AND ("algorithm") AND ("programming") AND ("beginner" OR "novice" OR "introductory course" OR "CS1" OR "apprentice" OR "student" OR "learner")
Web of Science	(learning) AND (self-assessment) AND (programming) AND (beginner OR novice OR introductory course OR CS1 OR apprentice OR student OR learner OR algorithm)
Scopus	(self-assessment) AND ("programming") AND ("beginner" OR "novice" OR "introductory course" OR "CS1" OR "apprentice" OR "student" OR "learner" OR "algorithm")
Science Direct	(self-assessment) AND (programming) AND (beginner OR novice OR introductory course OR CS1 OR apprentice OR student OR algorithm)

Fonte: Autoria Própria.

As bases selecionadas, de forma empírica, para a busca por estudos primários foram a *IEEE Xplore*, *ACM Digital Library*, *Web of Science*, *Scopus* e *ScienceDirect*.

De acordo com o Quadro 2, percebe-se algumas variações nas *strings* de busca previamente definidas. Isso se deve ao fato de algumas bases terem retornado artigos que não possuíam relação com o tema. Segundo Petersen, Vakkalanka e Kuzniarz (2015), o uso de *strings* de busca variadas não atrapalha os resultados da pesquisa de uma RSL, uma vez que resulta em uma maior variedade de resultados na área de pesquisa.

As buscas realizadas para encontrar os trabalhos primários foram realizadas considerando: (i) análise do título do trabalho, comparando os termos presentes na *string* utilizada ou até mesmo os sinônimos; e (ii) leitura do resumo da obra, identificando os termos relacionados à autoavaliação e programação introdutória.

Para melhorar o gerenciamento do processo, foi utilizada a StArt<sup>1</sup>, que é uma ferramenta de apoio ao processo de RSL, pois segue o modelo proposto por Kitchenham (FERREIRA, 2019). A StArt permite baixar informações relevantes dos trabalhos através das bases de busca que disponibilizam as informações em arquivos do tipo Bibtex. Ao

<sup>1</sup> Disponível em: <[http://lapes.dc.ufscar.br/tools/start\\_tool](http://lapes.dc.ufscar.br/tools/start_tool)>.

importar esse tipo de arquivo na StArt, ela coleta automaticamente os principais dados da obra, como título, autores, palavras-chave, local onde foi publicada, resumo, ano e tipo do artigo. Além disso, a StArt atribui uma pontuação à obra com base nas palavras-chave utilizadas na revisão sistemática em andamento.

### 3.3 CRITÉRIOS DE INCLUSÃO E EXCLUSÃO

Os critérios de seleção em uma RSL são definidos com a finalidade de buscar evidências nos estudos primários que estejam relacionadas com as questões de pesquisa (KITCHENHAM; CHARTERS, 2007). Considerando isso, critérios de inclusão e exclusão foram estabelecidos para identificar quais trabalhos deveriam ou não ser selecionados. O Quadro 3 apresenta os critérios de inclusão e exclusão que foram aplicados no período de novembro a dezembro de 2021.

Quadro 3: Critérios de inclusão e exclusão.

Critérios de Inclusão	
CI1	Estudos que relatam uso de técnicas de autoavaliação
CI2	Estudos que tenham data de publicação entre 2017 e 2021
CI3	Estudos que apresentem análises de técnicas de autoavaliação na aprendizagem de programação introdutória
CI4	Estudos publicados em inglês ou português
CI5	Estudos publicados em journals, revistas ou conferências
CI6	Estudos que fazem a medição da autoavaliação na aprendizagem de programação introdutória
Critérios de Exclusão	
CE1	Estudos repetidos
CE2	Estudos que não estejam disponíveis na íntegra
CE3	Estudos cujo foco não corresponde às questões de pesquisa
CE4	Estudos que não aplicam a autoavaliação na aprendizagem
CE5	Estudos que não possuem resumo
CE6	Estudos com menos de 5 páginas
CE7	Livros
CE8	Mapeamentos sistemático
CE9	Revisões sistemáticas

Fonte: Autoria Própria.

Vale ressaltar que os critérios são aplicados após a fase de buscas nas bases de dados com uso das *strings* e consequente seleção de artigos que possuam alguma relação com a abordagem da RSL. Inicialmente, os critérios de inclusão e exclusão foram aplicados durante a leitura das seções de resumo, introdução e conclusão, e, em alguns casos, metodologia (quando não havia clareza das informações dispostas no resumo, introdução e conclusão). Estudos que atenderam aos critérios de inclusão, foram selecionados para as próximas etapas do protocolo da pesquisa. Do mesmo modo, os trabalhos que atenderam aos critérios de exclusão foram descartados. Por fim, no que se

refere ao período de tempo dos estudos coletados para a pesquisa, foi definido como relevante e atual a faixa de 5 anos (2021-2017) devido a uma escassez de um levantamento do estado da arte, que apenas uma RSL consegue prover neste período de tempo e sobre esse tema.

### 3.4 AVALIAÇÃO DE QUALIDADE

Nessa etapa faz-se necessário o desenvolvimento de meios avaliativos da qualidade, que podem se dar através de perguntas elaboradas com base nas questões de pesquisa, com a finalidade de reforçar os critérios de inclusão e exclusão, possibilitando que estudos sejam descartados temporariamente, mesmo que tenham passado pela avaliação inicial de resumo, introdução e conclusão. Nesta pesquisa, utilizou-se um total de cinco (5) questões de avaliação de qualidade.

As questões elaboradas foram feitas de modo a permitir que uma nota fosse atribuída aos estudos por cada questão respondida completamente, parcialmente ou não respondida, sendo atribuída a pontuação de um ponto (1), meio ponto (0.5) e zero ponto (0), respectivamente. A nota máxima possível a ser alcançada por um estudo é de cinco pontos (5), caso responda corretamente todas as questões. O Quadro 4 apresenta as questões da avaliação de qualidade empregadas neste estudo.

Quadro 4: Questões da avaliação de qualidade.

<b>Questões da Avaliação de Qualidade</b>	
<b>QA1</b>	As ferramentas e métodos autoavaliativos utilizados foram bem definidos?
<b>QA2</b>	Os experimentos mencionados usando técnicas de autoavaliação foram realizados em um ambiente educacional introdutório de nível superior?
<b>QA3</b>	Está definido, de forma clara, quais propostas de autoavaliação obtiveram bons resultados?
<b>QA4</b>	Ao final do estudo, há uma discussão sobre os resultados obtidos acerca da metodologia/técnica de autoavaliação?
<b>QA5</b>	Está bem definido quais abordagens podem ser utilizadas para a autoavaliação na aprendizagem de programação introdutória?

Fonte: Autoria Própria.

Os trabalhos que obtiveram notas zero (0) na avaliação de qualidade foram separados em uma lista, ficando disponíveis para uma possível reavaliação e os demais seguiram para os próximos passos.

### 3.5 EXTRAÇÃO DE DADOS

A extração de dados é uma etapa do processo de RSL que consiste na coleta de dados que estão contidos nas obras primárias e que tem como objetivo auxiliar na

resolução das questões de pesquisas estabelecidas no protocolo. Os dados extraídos foram:

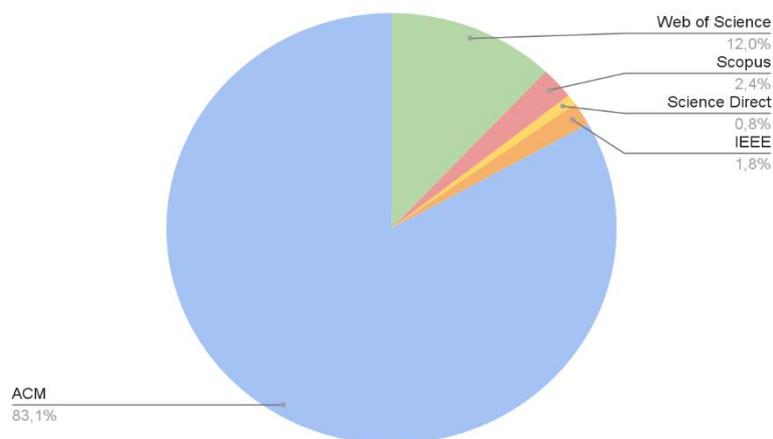
- Ano de publicação;
- Título;
- Idioma;
- Fonte;
- Autor(es);
- Técnicas de autoavaliação utilizadas;
- Desafios e limitações reportados pelos autores quanto à aplicação da(s) técnica(s);
- Abordagem de autoavaliação utilizada;
- Forma de medição da autoavaliação;
- Quais os benefícios da autoavaliação;
- Como a tecnologia ajuda a autoavaliação.

A ferramenta StArt foi utilizada como apoio nessa fase da RSL, pois ela é capaz de armazenar e coletar informações mais genéricas dos estudos de forma mais rápida e eficiente, além de organizar os estudos em listas, que podem ser facilmente consultadas.

#### 4 RESULTADOS E DISCUSSÕES

Nesta seção, são relatados e discutidos os resultados das análises feitas sobre os estudos primários. Com o uso das *strings* de busca, houve um volume considerável de estudos retornados pelas bases, conforme ilustrado no Gráfico 1.

Gráfico 1: Relação de bases e trabalhos encontrados.

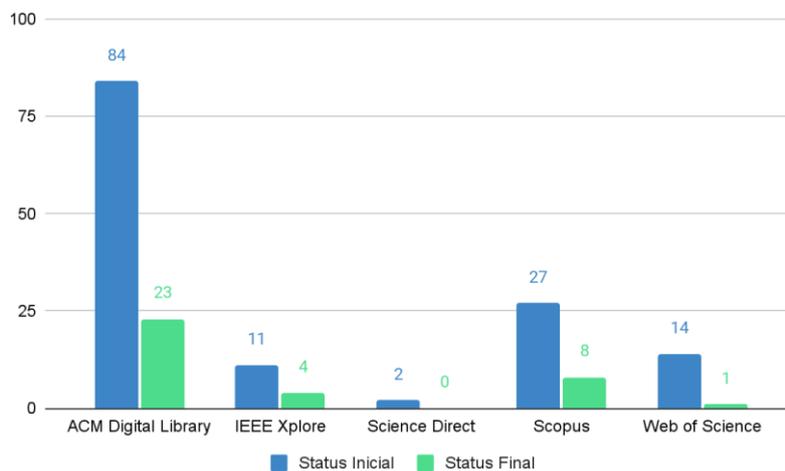


Fonte: Aatoria Própria.

Dos 2456 estudos retornados, 2318 foram descartados após leitura do título e resumo, pois os mesmos não possuíam relação com o tema de autoavaliação na aprendizagem de programação introdutória, não sendo categorizados como estudos primários, assim como definido na Subseção 3.2. Dessa forma, os demais 138 artigos, foram selecionados como estudos primários, pois possuíam alguma relação com o tema da RSL. Os critérios de inclusão e exclusão foram aplicados nos estudos primários selecionados na etapa anterior, tomando como base o que foi definido na Subseção 3.3. Portanto, dos 138 iniciais, 102 foram excluídos e 36 foram aceitos para seguir à avaliação de qualidade e análise de resultados. Toda a organização dos artigos nessa etapa foi feita com o auxílio da StArt, onde as obras aceitas, no final, foram devidamente catalogadas com a extração do título, ano de publicação, base e nota de qualidade, conforme mostra o Quadro 5, definido na Subseção 4.3.

O Gráfico 2 apresenta a quantidade inicial e final dos estudos selecionados mediante as etapas de busca e seleção dos estudos, análise com base nos critérios de inclusão e exclusão, bem como a avaliação da qualidade. Não foi necessário reavaliar nenhum dos trabalhos que foram separados em listas de exclusão, conforme sugerido na Subseção 3.4.

Gráfico 2: Comparação entre trabalhos classificados no início versus na última fase de seleção.



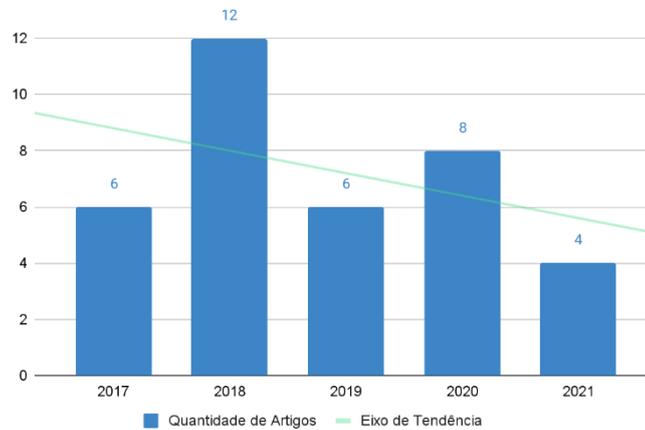
Fonte: Autoria Própria.

A seleção final resultou em 23 trabalhos da base *ACM Digital Library*, 4 trabalhos da base *IEEE Xplore*, nenhum trabalho da *Science Direct*, 8 trabalhos da base *Scopus* e 1 trabalho da base *Web of Science*.

#### 4.1 PERIODICIDADE DE PUBLICAÇÃO

O Gráfico 3 apresenta uma visão geral dos estudos selecionados em relação ao ano de publicação.

Gráfico 3: Quantidade de trabalhos selecionados por ano.



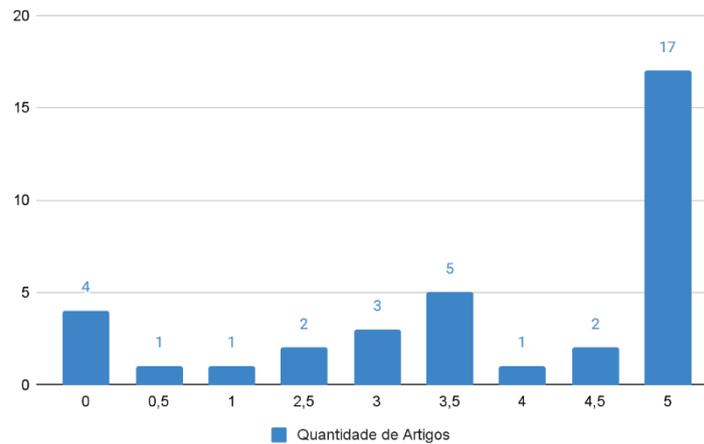
Fonte: Autoria Própria.

Conforme pode ser visto no Gráfico 3, verificou-se 12 estudos publicados no ano de 2018, seguido de 8 estudos no ano de 2020, 6 estudos tanto em 2017 quanto em 2019 e, por fim, 4 estudos publicados em 2021. Com isso, fica evidente que, ao longo dos anos, os estudos sofreram uma variação com tendência de diminuição a partir de 2018.

#### 4.2 AVALIAÇÃO DA QUALIDADE

Partindo das questões de qualidade definidas nesta RSL, a nota máxima que um estudo pode obter é de 5 pontos. Os estudos ilustrados no Gráfico 4 apresentam uma nota média de 3.76, diante dos critérios de avaliação adotados e um desvio padrão de 2.02 para mais ou para menos. Desse modo, julga-se uma média consideravelmente boa, dado que 47.2% dos estudos obtiveram nota máxima.

Gráfico 4: Notas da avaliação de qualidade dos estudos primários.



Fonte: Autoria Própria.

### 4.3 RESOLUÇÕES E DISCUSSÕES DAS QUESTÕES DE PESQUISA

Esta seção visa apresentar e discutir as resoluções das questões de pesquisa, definidas na Subseção 3.1. O Quadro 5 apresenta informações extraídas dos 36 artigos selecionados para responder às questões de pesquisa, dentre as quais, há o ID do artigo que será utilizado ao longo desta seção para referenciar os respectivos artigos.

Quadro 5: Informações dos artigos selecionados.

ID	Título	Ano	Base	Nota de Qualidade
E1	<i>The effects of bite-size distributed practices for programming novices</i>	2017	IEEE Xplore	5.0
E2	<i>An Algorithm for Assessment of Students Using Gamification</i>	2020	IEEE Xplore	3.5
E3	<i>Archimedes: Developing a Model of Cognition and Intelligent Learning System to Support Metacognition in Novice Programmers</i>	2020	IEEE Xplore	4.5
E4	<i>Quantitative Analytics in Exploring Self-Regulated Learning Behaviors: Effects of Study Persistence and Regularity</i>	2019	IEEE Xplore	4.5
E5	<i>Cs1 student grade prediction: Unconscious optimism vs insecurity?</i>	2021	Scopus	5.0
E6	<i>System-based ontology for assessing Learnerâ€™s programming practical works activities (S_ Onto_ ALPPWA)</i>	2021	Scopus	5.0
E7	<i>Why do CS1 Students Think They're Bad at Programming?: Investigating Self-efficacy and Self-assessments at Three Universities</i>	2020	Scopus	5.0
E8	<i>Automatic feedback provision in teaching computational science</i>	2020	Scopus	5.0
E9	<i>Advances in designing a student-centered learning process using cutting-edge methods, tools, and artificial intelligence: An e-learning platform</i>	2019	Scopus	5.0
E10	<i>Empirical study of multimedia learning object to enhance introductory programming learning</i>	2017	Scopus	5.0
E11	<i>The self-assessment in e-learning and personalized feedback education</i>	2017	Scopus	3.0

<b>E12</b>	<i>Metacognitive calibration when learning to program</i>	2017	Scopus	5.0
<b>E13</b>	<i>Effectiveness of Reflection on Programming Problem Solving Self-Assessments</i>	2018	Web of Science	5.0
<b>E14</b>	<i>Exploring the Value of Student Self-Evaluation in Introductory Programming</i>	2019	ACM Digital Library	5.0
<b>E15</b>	<i>On the Use of Semantic-Based AIG to Automatically Generate Programming Exercises</i>	2018	ACM Digital Library	5.0
<b>E16</b>	<i>Metacognitive Difficulties Faced by Novice Programmers in Automated Assessment Tools</i>	2018	ACM Digital Library	5.0
<b>E17</b>	<i>Self-Evaluation Interventions: Impact on Self-Efficacy and Performance in Introductory Programming</i>	2021	ACM Digital Library	5.0
<b>E18</b>	<i>Evaluation of Algorithms to Support Novice Programmer</i>	2018	ACM Digital Library	5.0
<b>E19</b>	<i>Developing Assessments to Determine Mastery of Programming Fundamentals</i>	2018	ACM Digital Library	0.0
<b>E20</b>	<i>Learning Journals in Creative Programming Assessments: Exposing Bugs, Issues, and Misconceptions</i>	2020	ACM Digital Library	5.0
<b>E21</b>	<i>Improved Mobile Robot Programming Performance through Real-Time Program Assessment</i>	2017	ACM Digital Library	0.0
<b>E22</b>	<i>Progression Trajectory-Based Student Modeling for Novice Block-Based Programming</i>	2021	ACM Digital Library	0.0
<b>E23</b>	<i>Adaptive Remediation for Novice Programmers through Personalized Prescriptive Quizzes</i>	2018	ACM Digital Library	4.0
<b>E24</b>	<i>Visualization, Assessment and Analytics in Data Structures Learning Modules</i>	2018	ACM Digital Library	1.0
<b>E25</b>	<i>Caf\{e}: Automatic Correction and Feedback of Programming Challenges for a CSI Course</i>	2020	ACM Digital Library	5.0
<b>E26</b>	<i>The Impact of Adding Textual Explanations to Next-Step Hints in a Novice Programming Environment</i>	2019	ACM Digital Library	3.5
<b>E27</b>	<i>MRS: Automated Assessment of Interactive Classroom Exercises</i>	2018	ACM Digital Library	5.0
<b>E28</b>	<i>Investigating Novices' In Situ Reflections on Their Programming Process</i>	2020	ACM Digital Library	3.5
<b>E29</b>	<i>ArTEMiS: An Automatic Assessment Management System for Interactive Learning</i>	2018	ACM Digital Library	3.0
<b>E30</b>	<i>ISnap: Towards Intelligent Tutoring in Novice Programming Environments</i>	2017	ACM Digital Library	3.0
<b>E31</b>	<i>An Evaluation of the Impact of Automated Programming Hints on Performance and Learning</i>	2019	ACM Digital Library	3.5

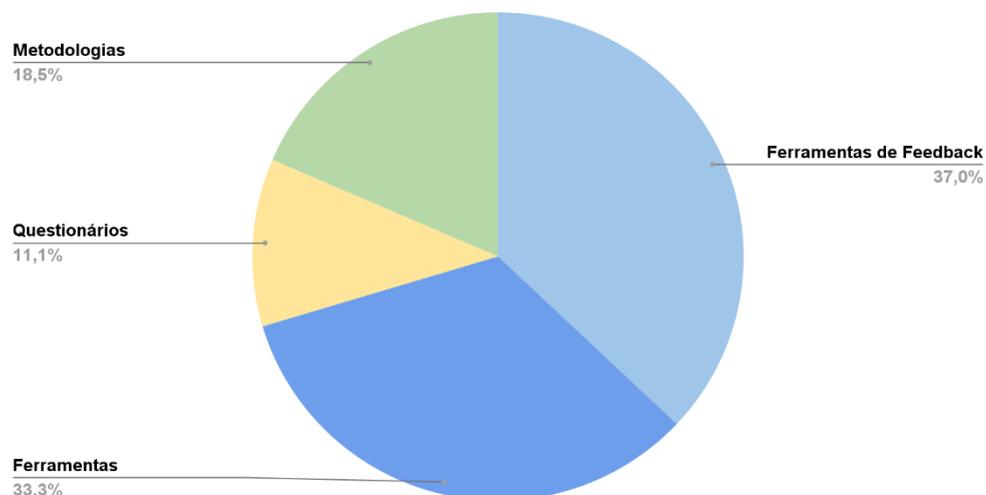
<b>E32</b>	<i>The Framework of a New Online Judge System for Programming Education</i>	2018	ACM Digital Library	0.5
<b>E33</b>	<i>Providing Meaningful Feedback for Autograding of Programming Assignments</i>	2018	ACM Digital Library	2.5
<b>E34</b>	<i>Automated Clustering and Program Repair for Introductory Programming Assignments</i>	2018	ACM Digital Library	2.5
<b>E35</b>	<i>Pair Teaching in Action</i>	2019	ACM Digital Library	0.0
<b>E36</b>	<i>Step Tutor: Supporting Students through Step-by-Step Example-Based Feedback</i>	2020	ACM Digital Library	3.5

Fonte: Autoria Própria.

A primeira questão de pesquisa refere-se a quais abordagens podem ser utilizadas para a autoavaliação na aprendizagem de programação introdutória. Dos trabalhos analisados, 75% utilizaram alguma abordagem de autoavaliação e os 25% restantes não utilizaram nenhuma abordagem de autoavaliação. Para melhorar a listagem dos tipos de abordagens, definiu-se 4 categorias: (i) ferramenta; (ii) ferramenta de *feedback* personalizado; (iii) questionário; e (iv) metodologia, conforme ilustrado no Gráfico 5.

Dos estudos que utilizaram alguma abordagem, 37% [E2, E8, E14, E15, E16, E23, E25, E27, E29, E34] foram classificados como ferramentas de *feedback* personalizado, o que correspondem a sistemas desenvolvidos pelos próprios autores, sistemas web, aplicações *mobile*, *quizzes*, avaliação de código e geração de dicas.

Gráfico 5: Abordagens utilizadas para medição de autoavaliação na aprendizagem de programação introdutória.



Fonte: Autoria Própria.

Por outro lado, 33.3% [E1, E4, E6, E9, E11, E12, E13, E24 e E26] fizeram uso de ferramentas para implementar a autoavaliação, onde, em sua maioria, elas tratavam-se de questionários *online*, sistemas de *e-learning* com formulários personalizados feitos pelos docentes, sendo que algumas apresentaram formas de interagir com blocos de código e estruturas de dados e até mesmo extensões de aplicativos *online* já existentes.

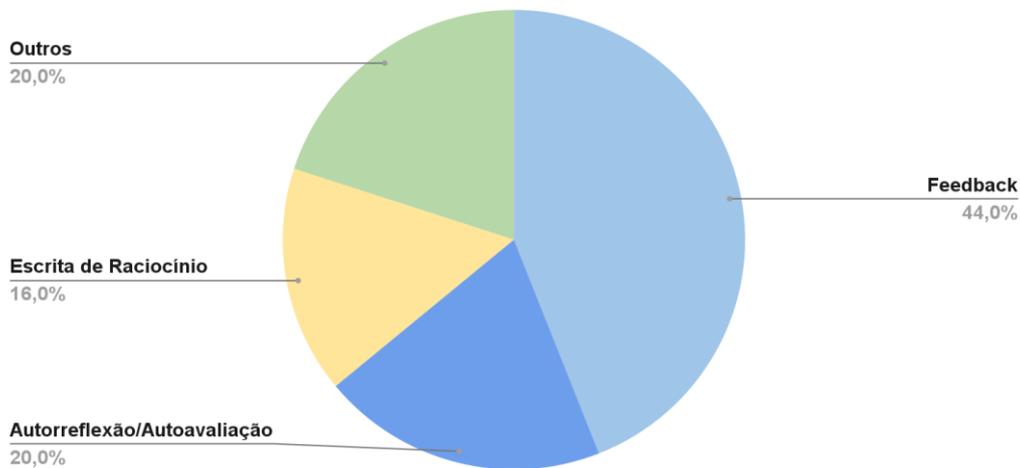
Os estudos classificados como metodologias de ensino corresponderam a 18.5% [E10, E18, E20, E28 e E33], isso significa que ao longo do período de aulas, os professores implementaram formas de aprendizagem diferentes, como decomposição de problemas, anotações diárias para posterior avaliação, *feedback* e incentivo da autorregulação.

Por fim, cerca de 11.1% dos estudos [E5, E7 e E17] foram classificados como questionários, que correspondem a implementação de abordagens autoavaliativas com o uso de perguntas feitas aos discentes, para a produção de análises, autorregulação, predição de notas e intervenções de autoavaliação na aprendizagem de programação.

A segunda questão da pesquisa busca investigar qual a técnica de autoavaliação é mais utilizada na aprendizagem de programação introdutória. Dos estudos selecionados, 69.4% utilizaram alguma técnica de autoavaliação e 30.6% não informaram nenhuma técnica autoavaliativa.

Para analisar melhor os estudos que utilizaram técnicas de autoavaliação, definiu-se as seguintes categorias de acordo com o objetivo principal da técnica utilizada: (i) *feedback*; (ii) autorreflexão/autoavaliação; (iii) escrita de raciocínio; e (iv) outros, conforme ilustrada no Gráfico 6.

Gráfico 6: Técnicas de autoavaliação mais utilizadas na aprendizagem de programação introdutória.



Fonte: Autoria Própria.

Dos estudos que utilizaram técnicas de autoavaliação, 44% [E1, E6, E8, E9, E15, E14, E23, E26, E25, E27, E29] usaram o *feedback* como principal técnica para implementar os conceitos e características da autoavaliação dos alunos e práticas de programação. Tais estudos destacam formas criativas, como *feedback* com base no histórico do aluno em suas tentativas ao resolver questões (E25), até formas de prover um *feedback* imediato e mais detalhado (E29) e clássicas análises de código submetido pelos alunos (E8). Dessa forma, esses estudos coincidem, em sua maioria, com os estudos cuja abordagem utilizou alguma ferramenta de *feedback*.

Pode-se mencionar também que 20% [E4, E5, E7, E13 e E17] optaram por formas mais explícitas de autoavaliação/autorregulação, que avaliam tanto performance do aluno, em relação a trabalhos que ele já havia feito e quais foram seus pontos altos e baixos (E17), quanto promoviam a autorregulação do aluno com o auxílio de ferramentas, onde o aluno deveria administrar seu esforço dedicado a cada atividade, instigando assim a metacognição, autoavaliação e planejamento estratégico dos alunos (E4). Por fim, pode-se mencionar que também usaram da predição de notas, como forma de o aluno autoavaliar seu conhecimento e de perceber a relação entre os que foram confiantes e os que não foram e seus resultados (E5).

Ainda se tem que 16% [E3, E16, E20, E28] dos estudos correspondem à classificação de escrita de raciocínio, onde a escrita comum foi fortemente usada como meio de aplicar a autoavaliação, com base em formas de reflexão do aluno. Pode-se mencionar os diários utilizados em (E20 e E28), que, de modo geral, correspondem à

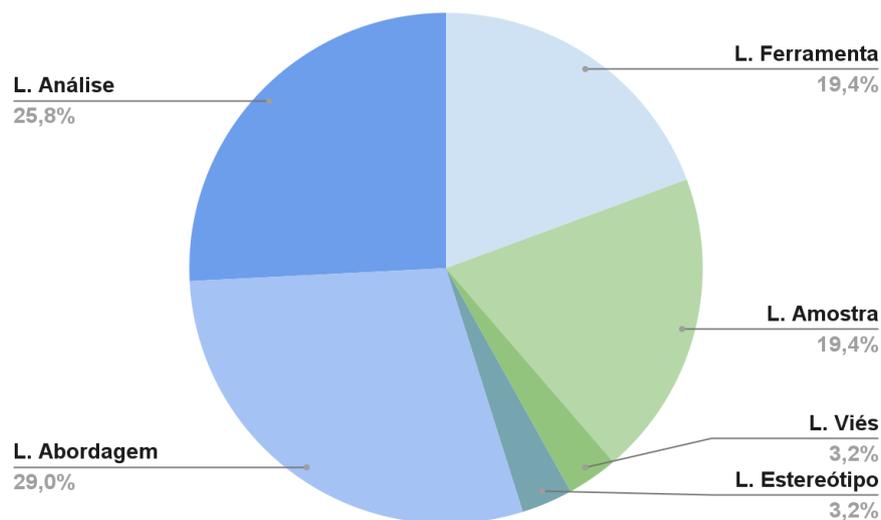
escrita do aluno durante a resolução de problemas ou estudo de um assunto, para que o próprio aluno possa refletir sobre o que estava fazendo.

Por fim, 20% [E2, E10, E11, E12, E18] dos estudos são classificados como outros, que possuem variadas técnicas de autoavaliação, quer seja por meio da gamificação como estímulo para participação do aluno e posterior autoavaliação (E2), ou por formas de aprendizado personalizado com o uso de sistemas *e-learning* associados ao ambiente de ensino tradicional (E11), ou até mesmo com a implementação dos chamados objetos de aprendizagem, em que o conhecimento adquirido em uma etapa serve como base para as próximas etapas (E10) e, por fim, o uso de pseudocódigo para a detecção de erros dos alunos em diferentes níveis da programação, provendo *feedback* significativo (E18).

A terceira questão da pesquisa busca investigar quais as principais limitações reportadas pelos autores na aprendizagem de programação introdutória. Dos estudos selecionados, 55.6% reportaram alguma limitação no processo de aprendizagem de programação introdutória e 44.4% não informaram nenhuma limitação.

Definiu-se seis categorias para as limitações reportadas: (i) limitação de ferramenta; (ii) limitação de amostra; (iii) limitação de estereótipo; (iv) limitação de análise; (v) limitação de viés; e (vi) limitação de abordagem/técnica, conforme apresentado no Gráfico 7. Ressalta-se que alguns estudos acabaram apresentaram mais de uma limitação.

Gráfico 7: Principais limitações reportadas pelos autores na aprendizagem de programação introdutória.



Fonte: Autoria Própria.

Das limitações desses estudos, pode-se mencionar a limitação de análise como uma das que mais ocorreram, com 25.8%. As obras que reportaram esses tipos de limitações são [E4, E7, E14, E17, E25, E26, E28 e E31] que, em sua maioria, ocorreram por problemas relacionadas à fase de análise dos resultados ou por detalhes que não permitiram uma conclusão precisa sobre o objeto de estudo ou que poderiam alterar os resultados esperados.

Destaca-se também a limitação de abordagem/técnica, com 29% das ocorrências. A lista de estudos que reportaram esse tipo de limitação inclui os seguintes trabalhos [E7, E10, E11, E12, E16, E17, E23, E30, E36]. Essa limitação se refere a problemas relacionados à abordagem sugerida pelos autores, cujo estudo não pôde ser concretizado da forma que os autores esperavam ou alguns resultados poderiam ser mais bem aproveitados, caso a abordagem/técnica contasse com as características pontuadas na seção de limitações da obra.

Já a limitação de amostra, correspondeu a 19.4% das ocorrências [E4, E16, E20, E28, E31, E36] e refere-se a problemas relacionados à quantidade de pessoas que participaram nos estudos, onde os autores não foram capazes de obter a quantidade necessária ou satisfatória para uma melhor análise. Também pode-se mencionar problemáticas associadas à segmentação e escolha dos alunos que iriam participar do estudo.

A limitação de ferramenta foi reportada em 19.4% das ocorrências [E1, E2, E3, E4, E6, E26]. Essa limitação acontece geralmente quando a ferramenta, software ou sistema utilizado pelos autores não supre todas as necessidades esperadas ou por apresentar a ausência de algum recurso ou funcionalidade que poderia ser útil para a pesquisa e até mesmo por apresentar falhas, *bugs* e outras questões relacionadas ao seu uso.

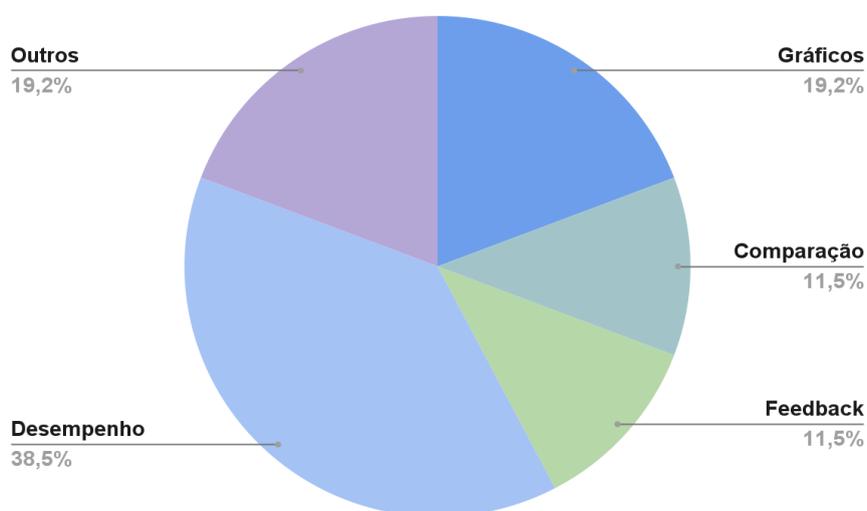
As demais limitações reportadas resumem-se em limitações de estereótipo [E4], onde informaram que tal pesquisa poderia gerar um estereótipo de “bom estudante” com a aplicação da autoavaliação e, no mesmo estudo, também ocorreu uma limitação de viés, pois devido a um alto foco na autoavaliação como atividade principal, os autores excluíram outras características importantes que podem acontecer em um contexto de sala de aula, criando um viés direcionado para a autoavaliação como única atividade, já que não foi levado em consideração quaisquer outra atividade metodológica.

A quarta questão de pesquisa tinha como objetivo investigar de que forma os estudos medem a autoavaliação em um cenário real de aprendizagem de programação

introdutória. Dos estudos analisados, 72.2% mediam os estudos de alguma forma e os 27.8% restantes não sinalizaram nenhum método para medição de autoavaliação em um cenário real de aprendizagem de programação introdutória.

Com a finalidade de dividir melhor a análise para responder a essa questão de pesquisa, os estudos foram classificados em 5 categorias, que representam suas características: (i) gráficos; (ii) comparação; (iii) *feedback*; (iv) desempenho; e (v) outros, conforme ilustrado no Gráfico 8.

Gráfico 8: Forma que os estudos medem a autoavaliação em um cenário real de aprendizagem de programação introdutória.



Fonte: Autoria Própria.

Dentre as formas de medir a autoavaliação, destaca-se a análise de desempenho, com 38.5% das ocorrências [E9, E10, E11, E12, E13, E14, E17, E20, E26, E27, E29]. A principal característica dessa forma de medição é que ela busca avaliar o desempenho final do aluno ao fazer as atividades do período letivo (como provas, projetos etc.) em relação ao que ele conseguiu desenvolver com a autoavaliação durante o estudo. Assim, se o desempenho for melhor do que o geral, significa um resultado positivo.

Em seguida, pode-se listar a classificação de medição baseada em gráficos, onde 19.2% [E1, E2, E4, E15, E23] dos estudos utilizaram desse recurso como forma de medir a autoavaliação. A principal característica dessa medição é a flexibilidade ou a possibilidade de diversas formas de plotagem do gráfico, o que permite aos autores terem diversas visões e uma variedade alta de análises que podem ser geradas.

Já 11.5% dos artigos [E5, E6, E7], utilizaram-se da comparação como forma de medir a autoavaliação. Essa métrica é interessante, pois permite aos autores avaliar um

cenário de antes e depois ou comparar os códigos desenvolvidos, e até fazer um comparativo de expectativa versus realidade.

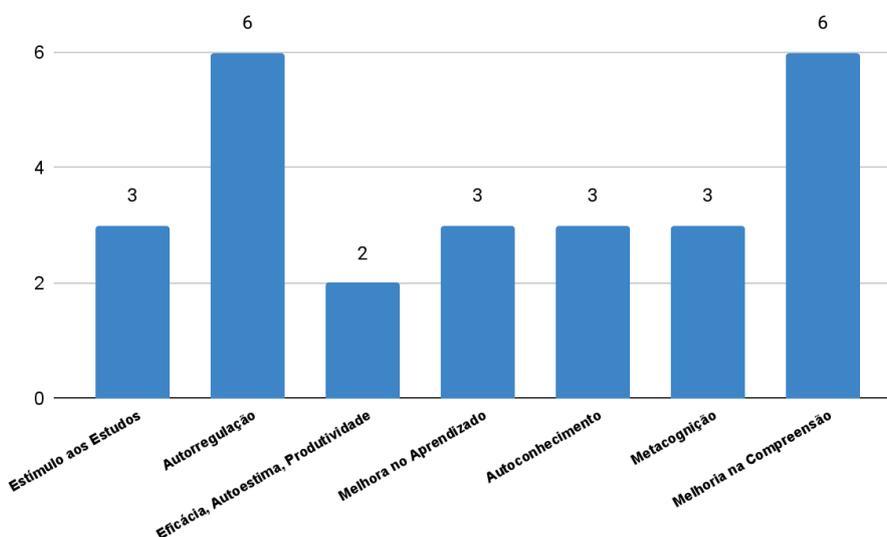
Pode-se mencionar ainda que 11.5% dos estudos [E8, E20, E31] optaram pela utilização de *feedback* como métrica para medir a autoavaliação. Esse *feedback* era provido pelo próprio aluno, onde ele respondia um questionário ou reportava diretamente aos autores sobre sua experiência e se ela foi positiva ou negativa, entre outros fatores que apoiassem a investigação da obra.

Por fim, 19.2% dos estudos ficaram na classificação de “outros” [E3, E8, E16, E18, E25 e E28]. Essa classificação diz respeito a uma variedade de outras formas de medição da autoavaliação, como, por exemplo, dimensões de análise (domínio de habilidade, coesão de *design*, caminho do desenvolvimento do programa, processo de idealização e estratégia organizacional), classificação de *statements* em um *journal* (alta cobertura, média cobertura e baixa cobertura), avaliação da capacidade de metacognição do estudante e uma análise de número de submissões por número de melhorias por número de pioras durante testes.

Para sanar a quinta questão de pesquisa, objetivou-se identificar quais os benefícios que a autoavaliação pode trazer para o aluno. Dos trabalhos analisados, 61.1% apresentaram quais benefícios a autoavaliação traz, enquanto 38.9% não trouxeram ou não avaliaram quais benefícios a autoavaliação pode trazer.

De forma geral, os estudos foram capazes de elencar diversos benefícios que a autoavaliação pode promover, conforme ilustrado no Gráfico 9.

Gráfico 9: Benefícios da autoavaliação reportados nos trabalhos.



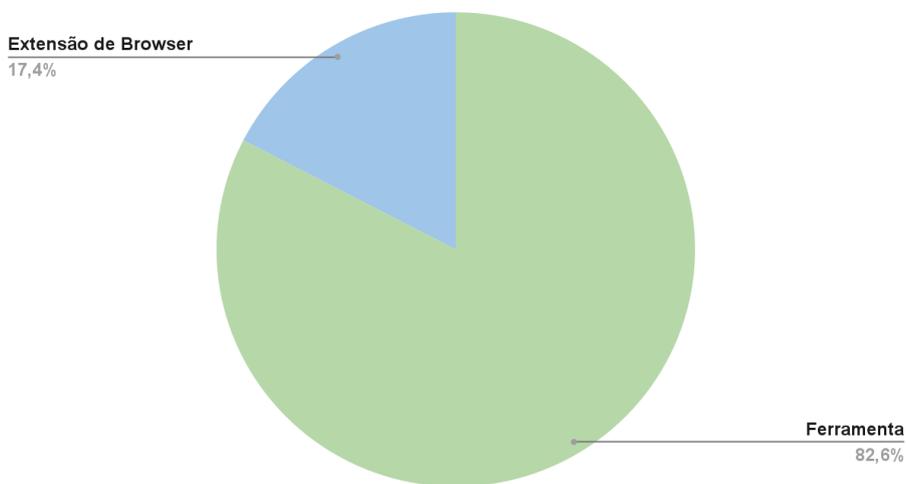
Fonte: Autoria Própria.

O benefício que todos os estudos possuem em comum é o de melhoria de performance dos alunos, o que significa que, após a técnica implementada com o uso de autoavaliação, os estudantes nos artigos que expuseram benefícios, contaram com um aumento de performance em comparação ao que conseguiam antes.

Já analisando os benefícios que não aconteceram em todos os estudos, pode-se mencionar o estímulo aos estudos [E2, E4, E8], autorregulação da aprendizagem [E3, E12, E13, E16, E26, E27], eficácia, autoestima e produtividade [E7, E23], melhoria no aprendizado de programação [E8, E10, E16], autoconhecimento [E9, E11, E12], metacognição [E12, E25, E16] e melhoria da compreensão do conteúdo apresentado [E6, E13, E14, E15, E17, E29].

Por fim, para responder a sexta e última questão de pesquisa, que se refere a como a tecnologia tem ajudado a autoavaliação no processo de aprendizagem de programação introdutória, foi levantado que 63.9% dos estudos mencionaram como a tecnologia provocou esse auxílio e 36.1% não mencionaram, conforme exposto no Gráfico 10.

Gráfico 10: Como a tecnologia tem ajudado a autoavaliação no processo de aprendizagem de programação introdutória.



Fonte: Autoria Própria.

Dos estudos que foram classificados com algum tipo de auxílio tecnológico, 82.6% deles utilizaram alguma ferramenta como forma de auxílio, desde sistemas de *quizzes*, a sistemas desenvolvidos pelos próprios autores com práticas e características da autoavaliação, sites, plataformas de *e-learning*, entre outros tipos de ferramentas. As obras que possuem essa característica são [E1, E2, E3, E4, E5, E6, E8, E9, E11, E12, E13, E14, E15, E16, E23, E24, E25, E27, E29].

Por outro lado, os 17.4% [E26, E30, E31, E36] dos estudos restantes utilizaram a extensão para *browsers* conhecida como *iSnap*, que fornece novas funcionalidades para uma plataforma já existente, a plataforma Snap!

## 5 TRABALHOS RELACIONADOS

Existem alguns trabalhos relacionados a esta RSL, como o de França e Tedesco (2015), Aureliano e Tedesco (2012) e Silva et al. (2015), que descobriram em seus estudos uma preferência dos autores por usarem/proporem ferramentas de software como forma de auxílio ao ensino-aprendizagem de programação e com foco no ensino superior.

Por outro lado, Rissi e Lucas (2019) notaram uma escassez de estudos que abordam procedimentos avaliativos relacionados à avaliação formativa. Já Medeiros, Falcão e Ramalho (2020), ao explorarem os desafios do ensino e aprendizagem de introdução à programação no Brasil, puderam apresentar estratégias, dificuldades e habilidades prévias que auxiliam no processo, tanto para professores quanto para alunos.

Por fim, Holanda, Freire e Coutinho (2019) conseguiram descobrir em sua pesquisa que mesmo as ferramentas propostas nas obras analisadas serem feitas para *desktop* e *mobile*, algumas acabaram por não terem continuação em seu desenvolvimento, o que afetou alguns alunos e professores que as utilizavam.

Esta RSL difere-se das demais discutidas nesta seção pelo intervalo de tempo adotado, isto é, por tratar de obras dos últimos 5 anos (2017 - 2021) e também pela relevância das cinco grandes bases de busca utilizadas como fonte de artigos, com *strings* de busca amplas que puderam retornar trabalhos capazes de sanar as questões relacionadas aos benefícios, limitações, técnicas, abordagens, abordagens de medição e como a tecnologia auxilia na autoavaliação na aprendizagem de programação introdutória.

## 6 CONSIDERAÇÕES FINAIS E TRABALHO FUTUROS

Nesta obra, foi conduzida uma RSL para analisar a autoavaliação na aprendizagem de programação introdutória, levando em consideração diversos aspectos relacionados em como a autoavaliação se aplica, buscando ainda identificar, principalmente, as técnicas e abordagens mais utilizadas, benefícios para os alunos, limitações e como a tecnologia auxilia nesse processo. Os estudos foram extraídos em um período de 5 anos, onde técnicas e abordagens de autoavaliação foram encontradas e os resultados foram discutidos de forma gráfica.

Por meio deste estudo, pode-se concluir que a técnica de autoavaliação, por meio de *feedback*, foi utilizada em 44% dos trabalhos, sendo a mais utilizada. Das abordagens utilizadas, as predominantes foram as ferramentas com *feedback* personalizado (37%), que proveram vários benefícios aos alunos, como a autorregulação e a melhoria na compreensão do conteúdo. A abordagem de medição da autoavaliação mais utilizada na aprendizagem de programação foi a performance dos estudantes (38.5%) e a forma como a tecnologia auxiliou esse processo foi, majoritariamente (82.6%), por meio de ferramentas. Por fim, a maior limitação reportada (29%) foi a limitação relacionada à abordagem utilizada.

Por fim, a presente RSL possui um propósito embrionário para um objetivo maior. Para obras futuras, pretende-se utilizar as informações apresentadas neste estudo como forma de desenvolver abordagens ou técnicas que gerem os benefícios levantados e sejam capazes de se sobressair sobre as limitações elencadas, o que deve fornecer formas de motivar os estudantes em cursos introdutórios de programação e melhorar o processo de ensino e aprendizagem e, conseqüentemente, melhorar os índices de aprovações.

### **AGRADECIMENTOS**

Agradecemos à Universidade Federal Rural do Semi-Árido (UFERSA) pela concessão de bolsas de iniciação científica.

## REFERÊNCIAS

- Aureliano, V. C. O. & Tedesco, P. C. A. R. (2012). Ensino-aprendizagem de Programação para Iniciantes: uma Revisão Sistemática da Literatura focada no SBIE e WIE. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, [S.l.]. ISSN 2316-6533. Disponível em: <<http://br-ie.org/pub/index.php/sbie/article/view/1718/1479>>. Acesso em: 09 ago. 2021. doi:<http://dx.doi.org/10.5753/cbie.sbie.2012.%p>.
- Azevedo, P. & Ferreira, M. J., (2014). "The use of response systems in the learning-teaching process", IX Iberian Conference on Information Systems and Technologies (CISTI). pp. 1-6, doi:10.1109/CISTI.2014.6877017.
- Batista, S. S. D. S. & Freire, E.; Delgado, D. M. (2020). Cursos Superiores de Tecnologia no contexto da internacionalização e da expansão da Educação Profissional e Tecnológica. *Série-Estudos*, Campo Grande, v. 25, n. 54, p. 193-221. Disponível em: <https://serieucdb.emnuvens.com.br/serie-estudos/article/view/1381/1105>. Acesso em: 20 jul. 2021.
- Bertelli, Y. & Gambarato, V. T. S. (2020). Importância Da Tecnologia Da Informação No Repensar Dos Negócios. In: Jornada Científica E Tecnológica Da Fatec De Botucatu, 9. São Paulo. *Anais da 9ª Jornada Científica e Tecnológica da Fatec de Botucatu*. São Paulo: Fatec de Botucatu. Disponível em: <http://www.jornacitec.fatecbt.edu.br/index.php/IXJTC/IXJTC/paper/viewFile/2342/2879>. Acesso em: 20 jul. 2021.
- Brizola, J. & Fantin, N. (2016). *Revisão da Literatura e Revisão Sistemática da Literatura*. Revista de Educação do Vale do Arinos, Juara, v. 3, n. 2, p. 23-29. Disponível em: <https://periodicos.unemat.br/index.php/relva/article/download/1738/1630>. Acesso em: 28 mai. 2021.
- Capellato, P. & Vasconcelos, L. V. B., Ranieri, M. G. A., Sachs, D. (2020). Peer and self-evaluation using active teaching method. *Research, Society and Development*, v. 9, n. 7. DOI: 10.33448/rsd-v9i7.3495. Disponível em: <https://rsdjournal.org/index.php/rsd/article/view/3495>. Acesso em: 25 jul. 2021
- Dermeval, D., Coelho, J. A. P. de M. & Bittencourt, Ig I. (2020). *Mapeamento Sistemático e Revisão Sistemática da Literatura em Informática na Educação*. In: JAQUES, Patrícia Augustin; SIQUEIRA; Sean; BITTENCOURT, Ig; PIMENTEL, Mariano. (Org.) Metodologia de Pesquisa Científica em Informática na Educação: Abordagem Quantitativa. Porto Alegre: SBC, 2020. (Série Metodologia de Pesquisa em Informática na Educação, v. 2) Disponível em: <https://metodologia.ceie-br.org/livro-2>.
- Falbo, R. A. *Mapeamento sistemático*. (2015). 25 p. Disponível em: <http://claudiaboeres.pbworks.com/w/file/fetch/133747116/Mapeamento%20Sistem%C3%A1tico%20-%20v1.0.pdf>. Acesso em: 28 mai. 2021.
- França, R. S. D. & Tedesco, P. C. D. A. R. (2014). Um modelo colaborativo para a aprendizagem do pensamento computacional aliado à autorregulação. In: III Congresso Brasileiro De Informática Na Educação (CBIE 2014), 2014, Dourados. *Anais do XXV Simpósio Brasileiro de Informática na Educação (SBIE 2014)*. Dourados: Sociedade Brasileira de Computação, 2014. p. 1133-1142, Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/3059>. Acesso em: 19 jun. 2021.

França, R. & Tedesco, P. Caracterizando a pesquisa sobre autoavaliação na aprendizagem de programação para iniciantes. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE), [S.l.], p. 549, out. 2015. ISSN 2316-6533. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/5312>> . Acesso em: 09 ago. 2021. doi:<http://dx.doi.org/10.5753/cbie.sbie.2015.549>.

Francisco, J. G. G. & Moraes, D. A. F. D. (2013). A Autoavaliação Como Ferramenta De Avaliação Formativa No Processo De Ensino e Aprendizagem. In: IV Seminário Internacional Sobre Profissionalização Docente, 2013, Curitiba. *Anais do XI Congresso Nacional de Educação*. Curitiba: Pontifícia Universidade Católica do Paraná, 2013. p. 14969-14983, Disponível em: [https://educere.bruc.com.br/CD2013/pdf/7225\\_4132.pdf](https://educere.bruc.com.br/CD2013/pdf/7225_4132.pdf). Acesso em: 20 jun. 2021.

Francisco, R., Ambrósio, A., Junior, C., & Fernandes, M. (2018). Juiz Online no ensino de CS1 - lições aprendidas e proposta de uma ferramenta. *Revista Brasileira de Informática na Educação*, 26(03), 163. doi:<http://dx.doi.org/10.5753/rbie.2018.26.03.163>

Giraffa, L. M. M & Moura, M. C. (2013) Evasão na Disciplina de Algoritmo e Programação: Um Estudo a Partir dos Fatores Intervenientes na Perspectiva do Aluno. In: *III Conferencia Sobre El Abandono En La Educación Superior*. 2013, Espanha, 2013. Disponível em: [https://repositorio.pucrs.br/dspace/bitstream/10923/8684/2/EVASAO\\_NA\\_DISCIPLINA\\_DE\\_ALGORITMO\\_E\\_PROGRAMACAO\\_UM\\_ESTUDO\\_A\\_PARTIR\\_DOS\\_FATORES\\_INTERVENIENTES\\_NA\\_PERSPECTIVA\\_DO\\_ALUNO.pdf](https://repositorio.pucrs.br/dspace/bitstream/10923/8684/2/EVASAO_NA_DISCIPLINA_DE_ALGORITMO_E_PROGRAMACAO_UM_ESTUDO_A_PARTIR_DOS_FATORES_INTERVENIENTES_NA_PERSPECTIVA_DO_ALUNO.pdf). Acesso em: 13 jun. 2021.

Holanda, W. D. D., Freire, L. D. P. & Coutinho, J. C. D. S. (2019). Estratégias de ensino-aprendizagem de programação introdutória no ensino superior: uma Revisão Sistemática da Literatura. *Renote*, Porto Alegre, v. 17, n. 1, p. 527-536, jul. 2019. Disponível em: <https://seer.ufrgs.br/renote/article/view/95905/53908>. Acesso em: 15 ago. 2021.

Kitchenham, B. & Charters, S. (2007) *Guidelines for performing Systematic Literature Reviews in Software Engineering*. EBSE Technical Report , Keele, p. 53, 9 Julho 2007. Disponível em: [https://www.elsevier.com/\\_\\_data/promis\\_misc/525444systematicreviewsguide.pdf](https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf). Acesso em: 26 mai. 2021.

Lopes, I. N. F. (2018). A Prática da Autoavaliação no Ensino Superior. *Id on Line Rev. Mult. Psic*, Juazeiro do Norte, v. 12, n. 39, 2018. Disponível em: <https://idonline.emnuvens.com.br/id/article/view/1033/0>. Acesso em: 25 jul. 2021.

Marxreiter, V. L. F., Bresolin, G. G. & Freire, P. de S. (2021). Autoavaliação: um olhar de inovação para a avaliação da aprendizagem das novas gerações. *P2P e Inovação*, v. 7, n. 2, p. 46-62, 2021. DOI: 10.21721/p2p.2021v7n2.p46-62. Disponível em: <http://revista.ibict.br/p2p/article/view/5633>. Acesso em: 28 jul. 2021.

Medeiros, R. P., Falcão, T. P. & Ramalho, G. L. (2020). Ensino e Aprendizagem de Introdução à Programação no Ensino Superior Brasileiro: Revisão Sistemática da Literatura. In: *Workshop Sobre Educação Em Computação (WEI)*, 28. , 2020, Cuiabá.

Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 186-190. ISSN 2595-6175. DOI: <https://doi.org/10.5753/wei.2020.11155>.

Nunziati, G. (1990). Pour construire un dispositif d'évaluation formatrice. *Cahiers Pédagogiques*, 280. (pp. 47-62).

Ngai, G. et al. (2010). On the implementation of self-assessment in an introductory programming course. In: SIGCSE BULL, 2010, New York. *Proceedings of the ACM SIGCSE Bulletin*. New York: Association for Computing Machinery, 2010. v. 41, p. 85-89, Disponível em: <https://dl.acm.org/doi/10.1145/1709424.1709453>. Acesso em: 20 jun. 2021.

Pacheco, J. A. Currículo: teoria e práxis. Porto: Porto Editora, v. 22, 1996.

Paiva, V. M. A. D. S. *Autoavaliação: uma proposta para a aprendizagem de língua inglesa no ensino médio*. 2017. 174 f. Tese (Doutorado) - Curso de Programa de Pós-graduação em Estudos da Linguagem, Centro de Ciências Humanas, Letras e Artes, Universidade Federal do Rio Grande do Norte, Natal, 2017. Disponível em: [https://repositorio.ufrn.br/bitstream/123456789/23495/1/VitoriaMariaAvelinoDaSilvaPaiva\\_TESE.pdf](https://repositorio.ufrn.br/bitstream/123456789/23495/1/VitoriaMariaAvelinoDaSilvaPaiva_TESE.pdf). Acesso em: 20 jul. 2021.

Pascoal, T. A., Brito, D. M. D. & Rêgo, T. G. D. (2015). Uma abordagem para a previsão de desempenho de alunos de Computação em disciplinas de programação. *TISE*, Santiago, v. 11, p. 454-458, dez. 2015. Trabalho apresentado no XX Congresso Internacional de Informática Educativa, Santiago, 2015. Disponível em: <http://www.tise.cl/volumen11/TISE2015/454-458.pdf>. Acesso em: 13 jun. 2021.

Petersen, K., Vakkalanka, S. & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information And Software Technology*, [s.l.], v. 64, p.1-18, ago. 2015. Elsevier BV.

Pinto, N. B. (1998). *O erro como estratégia didática no ensino da matemática elementar*. Tese (Doutorado em Didática) - Faculdade de Educação, Universidade de São Paulo, São Paulo, 1998. doi:10.11606/T.48.1998.tde-12022015-151819. Acesso em: 20 jun 2021.

Rezende, B. A. C. & Mesquita, V. D. S. (2017). O uso de gamificação no ensino: uma revisão sistemática da literatura. In: XVI Simpósio Brasileiro De Jogos e Entretenimento Digital, 2017, Curitiba. *Proceedings of SBGames*. Curitiba: Sociedade Brasileira de Computação, 2017. p. 1004-1007, Disponível em: <https://www.sbgames.org/sbgames2017/papers/CulturaShort/175052.pdf>. Acesso em: 13 jun. 2021.

Rissi, P. S. & Luccas, S. (2019). Estudos A Respeito Das Práticas Avaliativas Formativas: Avaliação Em Fases, Avaliação Por Pares E Autoavaliação. *Revista Conhecimento Online*, [S. l.], v. 3, p. 115-129, 2019. DOI:10.25112/rco.v3i0.1830. Disponível em: <https://periodicos.feevale.br/seer/index.php/revistaconhecimentoonline/article/view/1830>. Acesso em: 9 ago. 2021.

Santos, L. (2002) Auto-avaliação regulada: porquê, o quê e como? In: ABRANTES, P.; ARAÚJO, F. (Orgs.). *Avaliação das Aprendizagens: das concepções às práticas*. Lisboa: ME, 2002. p. 75-84, Disponível em: <https://repositorio.ul.pt/handle/10451/4884>. Acesso em: 20 jun 2021.

Schuster, M. E. (2008). *Mercado de Trabalho de Tecnologia da Informação: O Perfil dos Profissionais Demandado*. Orientador: Cinara Rosenfield. 2008. 59 f. TCC (Graduação) - Curso de Ciências Sociais, Departamento de Sociologia, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008. Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/17539/000718987.pdf?sequence>. acesso em: 24 jun. 2021.

Silva, M. A. D. (2016). *Autoavaliação: Uma Aliada No Processo De Construção Do Conhecimento Entre Alunos E Professores*. Orientador: Dr. Marcos Jorge. 2016. 135 f. Dissertação (Mestrado) - Curso de Faculdade de Ciências, Programa De Pós-graduação Em Docência Para A Educação Básica, Universidade Estadual Paulista, Bauru, 2016. Disponível em: [https://repositorio.unesp.br/bitstream/handle/11449/144616/silva\\_ma\\_me\\_bauru.pdf?sequence=4](https://repositorio.unesp.br/bitstream/handle/11449/144616/silva_ma_me_bauru.pdf?sequence=4). Acesso em: 25 jul. 2021.

Silva, T., Medeiros, T., Medeiros, H., Lopes, R., & Aranha, E. (2015). Ensino-aprendizagem de programação: uma revisão sistemática da literatura. *Revista Brasileira de Informática na Educação*, 23(01), 182. doi:<http://dx.doi.org/10.5753/rbie.2015.23.01.182>

Valadares, J. & Graça, M. (1998). *Avaliando para melhorar a aprendizagem*. Plátano Edições Técnicas.

Vieira, I. M. A. (2013). *A autoavaliação como instrumento de regulação da aprendizagem*. Orientador: Professora Doutora Lúcia Amante. 2013. 153 f. Dissertação (Mestrado) - Curso de Supervisão Pedagógica, Departamento de Educação e Ensino a Distância, Universidade Aberta, Lisboa, 2013. Disponível em: [https://drive.google.com/file/d/1Y6GuVM6EeRmQ8AgJ\\_xAx7fTOma0I7bok/view](https://drive.google.com/file/d/1Y6GuVM6EeRmQ8AgJ_xAx7fTOma0I7bok/view) Acesso em: 20 jun. 2021.

Vieira, D. B. M. M. (2017). *Conselho de Cooperação Educativa: A Participação dos Alunos na Regulação dos Processos de Aprendizagem*. Orientador: Professora Doutora Conceição Figueira . 2017. 168 f. Dissertação (Mestrado) - 1.º Ciclo do Ensino Básico e de Matemática e Ciências Naturais Escola Superior de Educação de Lisboa, Lisboa, 2017. Disponível em: [https://repositorio.ipl.pt/bitstream/10400.21/8513/1/Relat%c3%b3rio%20Final\\_Daniela%20Vieira.pdf](https://repositorio.ipl.pt/bitstream/10400.21/8513/1/Relat%c3%b3rio%20Final_Daniela%20Vieira.pdf). Acesso em: 20 jul. 2021.