

UNIVERSIDADE FEDERAL DE OURO PRETO  
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS  
DEPARTAMENTO DE COMPUTAÇÃO

MARIA EDUARDA DINIZ CAMARGOS  
Orientador: Prof. Dr. Rodrigo Geraldo Ribeiro

**UMA REVISÃO SISTEMÁTICA DA LITERATURA SOBRE FERRAMENTAS  
PARA CORREÇÃO DE TRABALHOS DE CURSOS INTRODUTÓRIOS DE  
PROGRAMAÇÃO DE COMPUTADORES**

Ouro Preto, MF  
2023

UNIVERSIDADE FEDERAL DE OURO PRETO  
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS  
DEPARTAMENTO DE COMPUTAÇÃO

MARIA EDUARDA DINIZ CAMARGOS

**UMA REVISÃO SISTEMÁTICA DA LITERATURA SOBRE FERRAMENTAS  
PARA CORREÇÃO DE TRABALHOS DE CURSOS INTRODUTÓRIOS DE  
PROGRAMAÇÃO DE COMPUTADORES**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

**Orientador:** Prof. Dr. Rodrigo Geraldo Ribeiro

**Coorientador:** Prof. Dr. Reinaldo Silva Fortes

Ouro Preto, MG  
2023

## SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

C172u Camargos, Maria Eduarda Diniz.

Uma revisão sistemática da literatura sobre ferramentas para correção de trabalhos de cursos introdutórios de programação de computadores. [manuscrito] / Maria Eduarda Diniz Camargos. - 2023. 44 f.

Orientador: Prof. Dr. Rodrigo Geraldo Ribeiro Ribeiro.

Coorientador: Prof. Dr. Reinaldo Silva Fortes Fortes.

Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Biológicas. Graduação em Ciência da Computação .

1. Linguagem de computadores. 2. Computação introdutória. 3. Ferramentas de correção automática. 4. Avaliação automática. I. Ribeiro, Rodrigo Geraldo Ribeiro. II. Fortes, Reinaldo Silva Fortes. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 004

Bibliotecário(a) Responsável: Luciana De Oliveira - SIAPE: 1.937.800



## FOLHA DE APROVAÇÃO

**Maria Eduarda Diniz Camargos**

**Uma revisão sistemática da literatura sobre ferramentas para correção de trabalhos de cursos introdutórios de programação de computadores.**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 23 de Março de 2023.

### Membros da banca

Rodrigo Geraldo Ribeiro (Orientador) - Doutor - Universidade Federal de Ouro Preto  
Reinaldo Silva Fortes (Coorientador) - Doutora - Universidade Federal de Ouro Preto  
Marcelo Luiz Silva (Examinador) - Mestre - Universidade Federal de Ouro Preto  
Leonardo Vieira dos Santos Reis (Examinador) - Doutor - Universidade Federal de Juiz de Fora

Rodrigo Geraldo Ribeiro, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 23/03/2023.



Documento assinado eletronicamente por **Rodrigo Geraldo Ribeiro, PROFESSOR 3 GRAU**, em 24/03/2023, às 07:35, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [http://sei.ufop.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0493048** e o código CRC **AE2A36B7**.

*Dedico esse trabalho aos meus pais que nunca mediram esforços para me ajudar a alcançar meus sonhos*

## **Agradecimentos**

Esta com certeza é uma das fases mais especiais da minha vida, e não posso deixar de agradecer meus pais por toda força, ânimo e coragem que me ofereceram para ter alcançado minha meta.

À Universidade e professores quero deixar uma palavra de gratidão por terem me fornecido tanto conhecimento de qualidade e apoio em todas as vezes que precisei.

À República Feitiço por ter sido meu lar e ponto de apoio durante toda minha graduação.

Por fim, um agradecimento especial à todas as pessoas que de uma alguma forma me ajudaram a acreditar em mim, sem elas não teria sido possível

Nunca se compare com ninguém neste mundo. Caso o faça, entenda que você estará insultando a si mesmo.

## Resumo

Aprender a programar é um desafio enfrentado pelos alunos na maioria dos cursos introdutórios de programação de computadores. Com um grande volume de exercícios práticos aplicados aos alunos, o uso de ferramentas automáticas de correção torna-se essencial para reduzir a carga de trabalho do professor e para apresentar resultados avaliativos mais rápidos sobre as atividades dos alunos. Neste sentido, o presente trabalho justifica-se pela necessidade de identificar as diferentes ferramentas disponíveis atualmente, seus recursos e limitações. A pesquisa foi realizada através de uma Revisão Sistemática da Literatura (RSL) e justifica-se pelo fato de ser uma metodologia de pesquisa que permite obter uma visão ampla sobre o tema de interesse, de forma precisa e reproduzível, e usada com sucesso na computação. O objetivo geral desta pesquisa é apresentar as estratégias utilizadas por diferentes ferramentas para correção de trabalhos de programação e seus impactos para a compreensão do aluno. Como objetivos específicos, o estudo almeja demonstrar diferentes ferramentas para correção de trabalhos de programação, bem como verificar suas características, linguagens de programação suportadas, relatos de uso e impacto no ensino de programação. Para a condução da coleta de dados, foi utilizado a ferramenta online Parsifal desenvolvida para apoiar pesquisadores na realização de revisões sistemáticas de literatura no contexto da Engenharia de Software. Como resultados, foram selecionados 108 artigos, dentre os quais oito escolhidos para estudo foram feitas análises qualitativas e quantitativas. Com base nas evidências fornecidas por este estudo, pode-se sugerir que os sistemas de avaliação automatizados, quando adequadamente integrados aos cursos de programação, podem gerar uma melhoria significativa no desempenho dos alunos. A partir dos dados levantados por esse trabalho, espera-se contribuir aos professores na escolha de uma ferramenta adequada às suas necessidades, ou direcionar para o desenvolvimento de sua própria ferramenta.

**Palavras-chave:** Computação introdutória. Avaliação automática. Ferramentas de correção automática. Programação de computadores.

## Abstract

Learning to program is a challenge faced by students in most introductory courses of computer programming. With a large volume of practical exercises applied to the students, the use of automatic correction tools becomes essential to reduce the teacher's workload and to present faster evaluation results on the students' activities. In this sense, the present work is justified by the need to identify the different tools currently available, their resources and limitations. The research was carried out through a Systematic Literature Review (SLR) and is justified by the fact that it is a research methodology that allows obtaining a broad view on the topic of interest, in a precise and reproducible way, and successfully used in computing. . The general objective of this research is to present the strategies used by different tools to correct programming assignments and their impact on student understanding. As specific objectives, the study aims to demonstrate different tools for correcting programming assignments, as well as verifying their characteristics, supported programming languages, usage reports and impact on programming teaching. To conduct the data collection, the Parsifal online tool developed to support researchers in carrying out systematic literature reviews in the context of Software Engineering was used. As a result, 108 articles were selected, among which eight chosen for study, qualitative and quantitative analysis were performed. Based on the evidence provided by this study, it can be suggested that automated assessment systems, when properly integrated into programming courses, can generate a significant improvement in student performance. From the data collected by this work, it is expected to contribute to teachers in choosing a suitable tool for their needs, or to direct them to the development of their own tool.

**Keywords:** Introductory computing. Automatic evaluation. Autocorrection tools. Computer programming.

## **Lista de fluxogramas**

Fluxograma 1 – Processo de triagem e seleção da RSL.....	9
Fluxograma 2 – Prisma 2020 para revisões sistemáticas.....	18

## **Lista de Tabelas**

Tabela 1 – Palavras chave e sinônimos em Inglês.....	12
Tabela 2 – Critérios de inclusão e exclusão de artigos.....	13
Tabela 3 – Relação dos artigos escolhidos para estudo.....	19
Tabela 4 – Características das ferramentas de correção.....	21
Tabela 5 – Feedback das ferramentas.....	22
Tabela 6 – Ferramentas e código fonte.....	23

## **Lista De Abreviaturas E Siglas**

ABNT	Associação Brasileira de Normas Técnicas
ABP	Aprendizagem Baseada em Projetos
DECOM	Departamento de Computação
PBL	Aprendizagem Baseada em Problemas
RSL	Revisão Sistemática da Literatura
TI	Tecnologia da Informação
UFOP	Universidade Federal de Ouro Preto

# Sumário

1	Introdução	1
1.1	Justificativa	3
1.2	Objetivos	3
1.2.1	Objetivo geral:	3
1.2.2	Objetivos específicos:	3
1.3	Organização do trabalho	4
2	Revisão bibliográfica	5
2.1	Trabalhos Relacionados	5
2.2	Fundamentação teórica	7
2.2.1	Revisão Sistemática da Literatura (RSL)	7
2.3	PLANEJAMENTO DA RSL	10
2.3.1	Definição do Problema	10
2.3.2	Definição dos objetivos	11
2.3.3	Escopo da pesquisa	11
2.3.4	Questões de pesquisa	11
2.3.5	Palavras-chave	12
2.3.5	Expressão de busca	12
2.3.8	Classificação dos dados	15
3.	Metodologia	16
3.1	Processo de busca e objetivo:	17
3.2	String de busca	17
3.3	Condução	17
3.4	Análise de resultados	17
4.	Resultados	18
4.1	Artigos escolhidos para estudo	19

4.2 Características das ferramentas de correção	20
4.3 Feedback das ferramentas	22
4.4 Ferramentas e código fonte	23
5. Considerações finais	25
5.1 Conclusão	25
5.2 Trabalhos e Atividades Futuros	26
Referências	27

## 1 Introdução

A disseminação dos sistemas computacionais em praticamente todos os setores da sociedade, faz com que a demanda de profissionais especializados no desenvolvimento de software cresça a cada ano.

De acordo com o artigo “Mercado de TI sofre com falta de profissionais” (G1, 2020), a formação de profissionais em universidades, escolas técnicas e cursos profissionalizantes não é suficiente para suprir as necessidades do mercado.

Entretanto, aprender a programar é um desafio enfrentado por alunos na maioria dos cursos introdutórios de programação de computadores. O aprendizado deste conteúdo envolve a resolução de muitos exercícios para que o aluno seja capaz de abstrair o padrão de raciocínio utilizado em diferentes situações e, então, possa aplicá-lo a outros problemas.

Segundo Arimoto (2019), os estudantes de cursos da área de computação encontram dificuldades e barreiras no processo de aprendizagem de programação de computadores, pois o conteúdo exige do aluno um alto grau de abstração, além de demandar tempo e esforço de aprendizagem. Adicionalmente, o estudo aponta a necessidade de abordagens que tornem o processo de aprendizado mais atrativo.

Porém, para que este estilo de aprendizado seja efetivo, deve haver retorno sobre a correção da solução construída por alunos. No contexto da Universidade Federal de Ouro Preto, o Departamento de Computação oferece de 20 a 22 turmas, com 22 vagas para a disciplina de Programação de Computadores I. Fornecer retorno a cada uma das atividades propostas para cada um dos alunos desta disciplina é praticamente impossível.

Diante da importância do ensino de programação e das dificuldades em prover retorno às atividades propostas, faz-se necessária a utilização de ferramentas que realizem a correção automática de exercícios de programação. O uso deste tipo de recurso permite que os alunos obtenham rapidamente um parecer sobre a correção de suas respostas, o que pode direcionar o estudo do conteúdo. De acordo com Brito (2019), o alto nível de abstração exigido nessas disciplinas pode ser causador de desmotivação entre os alunos, resultando em uma alta taxa de reprovação ou desistência do curso.

Porém, existem diversas propostas de ferramentas possuindo os mais diferentes recursos para a correção automática de exercícios de programação. Conforme menciona Kumar (2011), para que esse quadro melhore, muitas ferramentas de TIC (Tecnologia da Informação e Comunicação), têm sido sugeridas com o objetivo de auxiliar o professor no

ensino de programação, como a utilização dos Ambientes Virtuais de Aprendizagem (AVA), que tem como forte representante o Moodle.

Outra ferramenta muito usada para a correção automática de trabalhos de programação é o Run.codes, um sistema que corrige exercícios de programação em poucos segundos. A ferramenta possibilita que o aluno cadastre o trabalho de programação e, em segundos, ele é corrigido. Dessa forma o professor consegue gerir, com eficácia, o trabalho realizado (FAPESP, 2015).

No que se refere à avaliação de atividades de programação, ou seja, as atividades práticas de codificação, pode-se citar alguns sistemas como, por exemplo, os sistemas de Juízes Online. Estes sistemas são utilizados em maratonas de programação, com um vasto repositório de questões e um método de avaliação automática de código fonte, gerando respostas como: errado, certo, erro de compilação e erro em tempo de execução (ZHIGANG et al., 2001).

Um exemplo de repositório que disponibiliza os problemas computacionais utilizados em competições de programação é o SPOJ Brasil. Embora seu objetivo seja o de prover treinamento para maratonas, é também utilizado para aprendizado e treinamento de programação (SPOJ BRASIL, 2023).

Outra plataforma de programação competitiva é a Beecrowd, que oferece mais de 2.000 testes analíticos e de codificação disponível em mais de 25 linguagens de programação (BEECROWD, 2023).

Nas seções a seguir, serão apresentadas a justificativa desse estudo, os objetivos geral e específicos e como o trabalho foi organizado. Na sequência serão apresentados alguns trabalhos relacionados ao tema proposto, seguido de conceito e fundamentos da Revisão Sistemática da Literatura e todo seu mecanismo de busca e seleção de artigos para estudo. Finalmente, serão apresentados os resultados através de tabelas e discussão através de confrontos com algumas teorias fundamentadas por autores renomados, seguida de considerações finais e trabalhos futuros.

## **1.1 Justificativa**

A primeira disciplina de programação é fundamental em cursos de computação. Este conteúdo envolve habilidades como criatividade, raciocínio abstrato e estratégias para resolução de problemas. Adicionalmente, estudantes devem expressar suas soluções seguindo

as regras estritas impostas por uma linguagem de programação, o que adiciona um nível de dificuldade à aprendizagem.

Neste sentido, o presente trabalho justifica-se pela necessidade de identificar as diferentes ferramentas disponíveis atualmente, seus recursos e limitações. A partir dos dados levantados por esse trabalho, espera-se ajudar professores na escolha de uma ferramenta adequada às suas necessidades. Outro público que pode se beneficiar dos resultados deste trabalho são pesquisadores da área de informática na educação que podem propor melhorias ou mesmo novas ferramentas para a correção de exercícios de programação.

## **1.2 Objetivos**

### **1.2.1 Objetivo geral:**

Selecionar e analisar trabalhos de pesquisa, por meio de uma revisão sistemática da literatura, utilizando o protocolo PRISMA (SOUZA; BATISTA; BARBOSA, 2014), a fim de apresentar o estado da arte relacionado ao desenvolvimento e utilização de ferramentas para correção automática de exercícios de programação, bem como avaliar o impacto de sua utilização.

Visando fornecer critérios para a escolha deste tipo de ferramenta, o presente trabalho busca responder os seguintes questionamentos: “É possível personalizar os critérios para correção de um exercício? A geração de casos de teste é automatizada ou deve ser criada quando da construção do exercício? Qual tipo de retorno a ferramenta fornece para o aluno?” Essas são algumas perguntas que este trabalho almeja responder de forma completa ou parcial utilizando uma revisão sistemática da literatura (RSL)

### **1.2.2 Objetivos específicos:**

O presente trabalho almeja os seguintes objetivos específicos:

1. Identificar diferentes ferramentas para correção de trabalhos de programação;
2. Verificar as características e quais informações diferentes ferramentas apresentam após a análise da solução de uma questão;
3. Identificar as linguagens de programação suportadas pelas ferramentas consideradas;

4. Identificar relatos de uso de ferramentas de correção automática e seu impacto no ensino de programação.

### **1.3 Organização do trabalho**

O restante deste trabalho está organizado como se segue. O Capítulo 2 apresenta o referencial teórico para o desenvolvimento desta pesquisa. No Capítulo 3 é apresentada a metodologia para alcançar os objetivos deste estudo. Os resultados obtidos serão discutidos no Capítulo 4. Finalmente, o Capítulo 5.1 apresenta considerações finais e direções para possíveis trabalhos futuros.

## **2 Revisão bibliográfica**

No que tange à importância do uso de ferramentas para correção de trabalhos de cursos introdutórios de programação de computadores, várias abordagens foram propostas. Na Seção 2.1 serão apresentados os trabalhos relacionados, demonstrando algumas ferramentas para correção de trabalhos de programação e sua aplicabilidade. Na Seção 2.2 sobre a Revisão Sistemática da Literatura (RSL) e suas etapas. Na Seção 2.3 sobre o planejamento da RSL.

### **2.1 Trabalhos Relacionados**

Em Costa, (2020) a proposta da revisão sistemática compreendia o uso de tutores de programação inteligente na produção de feedback para estudantes em tarefas de programação. Identificou-se, como resultado, que o uso de tutores de programação traz impactos positivos nos cursos introdutórios de programação.

No estudo de Richter (2019), foi feita uma RSL buscando investigar e identificar as principais estratégias e ferramentas tecnológicas utilizadas para mediar o ensino da programação orientada a objetos. Foram analisados 136 artigos, dentre os quais apontaram que os softwares Alice e Scratch foram os mais indicados e as abordagens para o ensino POO mais usadas foram a Aprendizagem Baseada em Problemas (PBL) e Aprendizagem Baseada em Projetos (ABP).

Conforme Holanda (2019), em uma revisão sistemática da literatura feita no período de 2014 a 2018, que visou compreender sobre as estratégias adotadas para o ensino de programação introdutória em cursos superiores, demonstraram como resultados que os jogos digitais, bem como softwares educativos e metodologias específicas estão sendo utilizadas para minimizar a evasão de alunos nessas disciplinas.

Nesse mesmo viés, com base em RSL, o estudo de Ferri (2016), constatou através de uma metassíntese, que o ensino de programação básica, através das ferramentas SuperLogo, Mocromundos e Scratch, contribuíram significativamente para o conhecimento, potencializando os resultados na aprendizagem.

No estudo de Reis (2015), foi apresentado ferramentas de software para o ensino de programação que propõe jogos e uso da robótica, como Alice, Lego mind-storms NXT, Progest, Game Maker, Scratch. Os autores afirmam que a utilização de jogos digitais e

robótica para o ensino-aprendizagem de programação melhoram o rendimento dos alunos em disciplinas introdutórias de programação.

Em outro estudo de avaliação comparativa entre as linguagens Rust e C, para o ensino de programação, o autor Fukuda (2021), concluiu que Rust, uma linguagem de programação compilada multi-paradigma, desenvolvida pela Mozilla, é uma alternativa viável para o ensino de programação, pois possui um ambiente de desenvolvimento no qual todas as ferramentas são instaladas inicialmente, enquanto que a linguagem C, se mostrou boa, porém para programação em nível mais próximo ao hardware, necessitando de um IDE para facilitar a elaboração de programas.

As principais ferramentas apontadas nos estudos de Brito (2010), para facilitar o ensino-aprendizagem no curso de programação, foram o CodeBench (juiz online), que proporciona aos alunos um conjunto de ferramentas pedagógicas de estímulo ao aprendizado, com obtenção de feedback sintático e semântico. O URI Online Judge (juiz online), que estimula o aprendizado focalizado em conceitos de algoritmos e programação de computadores. E o The Huxley (juiz online), uma plataforma web gratuita com consulta de tutores online para dicas e orientações, com uma base sólida de exercícios para o aprendizado de algoritmos e programação de computadores.

No estudo de Zanetti e Oliveira (2015), foram utilizados o S4A, o Scratch 4 Arduino, e um robô, que recebia comandos através da plataforma Arduino, com o objetivo de encontrar soluções para os alunos com dificuldade na aprendizagem de programação. Este estudo foi feito através de oficina e prática, tendo como resultado positivo de que o método aplicado ajuda na composição da solução desenvolvida através do aluno. De acordo com os autores, a robótica pedagógica auxilia na abstração empírica, e permite que o aluno extraia informações mais concretas, tanto do objeto, quanto das ações sobre ele.

No trabalho de Raab, Zanchett e Vahldick (2015), a experiência com jogos como o Code Combat, trouxe uma maior facilidade de aprendizado entre os alunos, utilizando mecânicas simplificadas para a condução do aluno, com foco na resolução dos desafios de lógica de programação propostos no jogo. Como resultado, os autores perceberam que os alunos obtiveram um bom nível de aproveitamento e aprendizado para o curso introdutório de programação de computadores.

Na seção a seguir será apresentada toda a metodologia da Revisão Sistemática da Literatura, demonstrando as etapas do processo de busca, tratamento e escolha dos artigos que irão compor o estudo para elaboração da RSL.

## 2.2 Fundamentação teórica

A Revisão Sistemática da Literatura (RSL) permite obter uma visão ampla sobre um determinado tópico de interesse de forma precisa e reproduzível. A RSL surgiu no final da década de 70. O objetivo da RSL é buscar melhor qualidade das publicações científicas na área de medicina, sendo propagada para outras áreas de estudo.

Na computação, revisões sistemáticas têm sido usadas com sucesso na engenharia de softwares. A técnica de RSL foi introduzida na comunidade de ES pelo trabalho de Kitchenham.

Nas subseções a seguir, serão apresentadas as etapas de uma RSL, conforme ilustrado no Fluxograma 1.

Fluxograma 1 – Etapas da RSL



Fonte: Kitchenham (2004)

### 2.2.1 Revisão Sistemática da Literatura (RSL)

A RSL visa obter evidências para uma questão de pesquisa específica e envolve uma pesquisa reprodutível e completa da literatura e uma avaliação crítica dos estudos elegíveis. Isso é diferente das revisões narrativas, que permitem que os autores destaquem os achados de estudos selecionados (MAFRA e TRAVASSOS, 2006).

Como tal, a RSL é um estudo em que a população é oriunda de artigos já publicados, ser feita de forma objetiva, rigorosa e meticulosa do ponto de vista qualitativo e quantitativo. Ela compila as informações geradas por investigações acerca de um assunto específico, que às vezes é avaliado matematicamente com uma meta-análise, mantendo o efeito individual de cada estudo incluído (KITCHENHAM; CHARTERS, 2007).

Dessa forma, o resultado de cada estudo pode ser determinado no cálculo do efeito combinado, conforme as evidências geradas são sintetizadas. Estratégias são usadas para limitar vieses e erros aleatórios. Estes incluem pesquisas exaustivas de todos os artigos relevantes, critérios de seleção reprodutíveis, avaliação do desenho, características do estudo e síntese e interpretação dos resultados. Ao final, esses resultados são usados para desenvolver conclusões que resumem os efeitos comparados de diferentes intervenções (NICHOLSON et al, 2017).

A condução de uma RSL começa com uma questão de investigação, um protocolo com identificação clara de objetivos, resultados e um plano bem definido para análise e relato de resultados. É aconselhável registrar protocolos em um banco de dados disponível publicamente, permitindo transparência e aumentando a confiança entre os leitores nos resultados do estudo (KITCHENHAM e CHARTES, 2007).

Próximas subseções detalham cada uma das etapas de uma RSL.

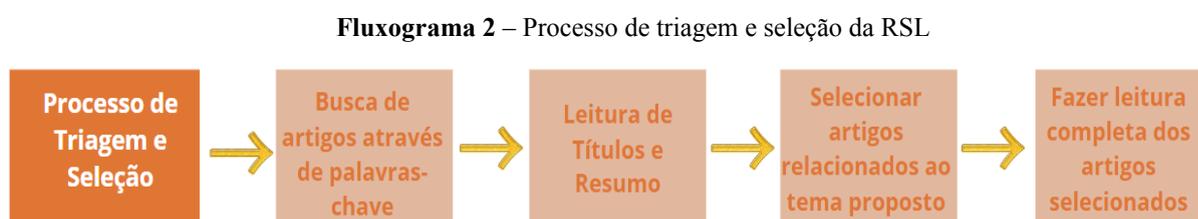
### **2.2.1.1 Formular uma pergunta de investigação**

Uma pergunta específica e claramente pensada é o primeiro passo para a conclusão de uma RSL. A pergunta geralmente é enquadrada no formato de intervenção, população, comparador e resultados e influencia o escopo da revisão. A decisão sobre o tipo de desenho de estudo a ser incluído é feita com base na questão da revisão (WRIGHT et al, 2007). A estratégia de busca de uma RSL é baseada na pergunta de pesquisa e pode variar um pouco com base nas práticas de indexação específicas do banco de dados. Dessa forma, as estratégias de pesquisa de literatura normalmente usam uma combinação de termos de cabeçalho e palavras-chave de texto livre (KITCHENHAM; CHARTERS, 2007).

Segundo Wright et al (2007), pode-se pesquisar no campo título e resumo das citações por um termo específico de texto livre. Ao combinar os resultados dos dois conjuntos de pesquisa, um conjunto maior de registros potencialmente relevantes é recuperado. Um equilíbrio entre sensibilidade e especificidade, onde se recupera estudos mais relevantes, é necessário para alcançar uma busca prática e abrangente. Após a remoção das duplicidades, é compilada uma lista final de citações (títulos e resumos) de todas as bases de dados revisadas.

### 2.2.1.2 Seleção de estudos relevantes

Com base em critérios de seleção, conforme descrito no Fluxograma 2, os artigos obtidos por meio de busca na literatura passam por um processo de triagem e seleção em duas etapas. A primeira é realizada em títulos e resumos em que se adota uma estratégia mais sensível para excluir claramente artigos não relacionados. Na segunda etapa, todos os estudos relevantes ao tema são pesquisados em texto completo para concluir a elegibilidade (MAFRA, 2005).



Fonte: Mafra (2005).

De acordo com Mafra (2005), em ambos os estágios deve-se realizar uma revisão minuciosa para garantir a confiabilidade. As discrepâncias na seleção são resolvidas por arbitragem. Exercícios de consenso entre revisores no início de cada etapa são aconselháveis. Ao final desse processo, obtém-se uma lista final de artigos a serem incluídos na RSL.

### 2.2.1.3 Extração de dados

Esta etapa envolve a extração de dados de estudos individuais necessários para a análise e relatório da revisão. Esses dados incluem, normalmente, características do desenho do estudo, informações demográficas dos participantes, itens de risco de viés e resultados de acompanhamento e estudo.

Semelhante à triagem, esse processo é realizado por pares de revisores, de forma independente, para evitar erros na aquisição de dados. A organização clara e o registro dos dados extraídos são necessários para facilitar uma revisão. Itens de dados ausentes ou pouco claros devem ser solicitados aos autores dos estudos e, se não estiverem disponíveis, devem ser claramente documentados (KITCHENHAM e CHARTERS, 2007).

#### **2.2.1.4 Avaliação da qualidade**

A qualidade do relatório de um estudo considera até que ponto os aspectos do desenho e da conduta de um estudo podem proteger contra o desvio da verdade, que são os erros sistemáticos. O processo de avaliação da qualidade é realizado durante a fase de extração de dados e garante que estudos futuros tenham uma relevância para a RSL. Para isso, várias ferramentas estão disponíveis para avaliação da qualidade metodológica dos estudos incluídos. Por exemplo a AMSTAR, construída a partir da análise e da atualização de outros instrumentos, empiricamente validado. Outro exemplo de ferramentas são o *checklists and guidelines*, que identificam, analisam e sintetizam os dados, ocasionando uma variabilidade na maneira de se conduzir uma RSL. A avaliação da qualidade é necessária para interpretar os resultados e classificar a força da evidência (WRIGHT et al, 2007)

#### **2.2.1.5 Síntese de dados (metanálise)**

Os aspectos estatísticos de uma RSL são denominados de meta-análise. Uma das premissas para a execução deste tipo de estudo é que o delineamento das pesquisas dos estudos primários seja homogêneo, por exemplo, efeitos medidos usando métricas similares, design de experimentos semelhantes, etc.

Para os estudos de meta-análise, a extração dos dados referentes ao delineamento da pesquisa e aos resultados encontrados é crucial para seu sucesso. A coleta destes dados numéricos constitui-se em um pré-requisito para a condução da meta-análise, pois usa técnicas estatísticas que integram os resultados dos estudos primários incluídos na revisão sistemática. (KITCHENHAM e CHARTERS, 2007).

#### **2.2.1.6 Relatório de resultados**

Nessa última etapa são apresentados e sistematizados os resultados obtidos com a pesquisa. Deve-se categorizar as informações extraídas dos artigos pesquisados. Planeja-se então a estratégia de disseminação da pesquisa, desenvolvimento e avaliação de recomendações usadas para avaliar e relatar a qualidade das evidências para cada resultado encontrado (MAFRA, 2005).

## **2.3 PLANEJAMENTO DA RSL**

Essa etapa compreende o planejamento detalhado, com orientação de cada fase do estudo, o tema, objetivos e escopo até chegar a análise dos dados, em que resultou na produção do protocolo de pesquisa.

### **2.3.1 Definição do Problema**

A definição do problema de pesquisa deve ser formulada em função do foco e do objetivo da revisão. No contexto da computação, pesquisadores em Engenharia de Software usam estas diretrizes como base para definir os tipos de problemas de pesquisa que podem ser formuladas dentro do tema escolhido e do tipo exploratória, relacionais, causais e de design.

### **2.3.2 Definição dos objetivos**

A definição de objetivos é um passo crucial na revisão sistemática da literatura. Os objetivos devem ser bem definidos e orientar todo o processo da revisão. Eles devem ser específicos e mensuráveis, além de estar alinhados com a questão de pesquisa central.

### **2.3.3 Escopo da pesquisa**

O escopo da pesquisa visa mapear a literatura sobre um determinado tópico ou área de pesquisa para identificar conceitos-chave e possíveis lacunas e fornecer uma visão descritiva dos estudos revisados.

#### **2.3.3.1 Idiomas**

Idiomas em revisão sistemática referem-se ao processo de revisão de uma literatura que está disponível em idiomas diferentes. Além disso, é importante salientar que este processo é feito de forma sistemática. Os idiomas em revisão sistemática são importantes porque há muita pesquisa científica que é realizada em diferentes idiomas, especialmente em

países diferentes. Como resultado, essa pesquisa pode não ser acessível ou utilizável por pesquisadores de outros países, especialmente aqueles que não falam esses idiomas.

### ***2.3.3.2 Intervalo de tempo***

O intervalo de tempo em revisões sistemáticas se refere ao período em que a pesquisa foi realizada e inclui a literatura disponível naquele intervalo. É importante definir esse intervalo ao realizar uma revisão sistemática, pois isso afeta a validade e a relevância dos resultados.

### **2.3.4 Questões de pesquisa**

A pergunta de pesquisa em RSL (Revisão Sistemática da Literatura) é uma das etapas mais importantes e críticas deste processo. Essa pergunta é o guia que orientará todo o estudo de revisão sistemática, e sua formulação deve ser cuidadosa e precisa para garantir que todos os resultados encontrados sejam relevantes e adequados ao objetivo do estudo.

#### ***2.3.4.1 Questões primárias***

As questões primárias em revisão sistemática são perguntas específicas que norteiam a busca pelas evidências relevantes para responder a uma questão de pesquisa específica. Essas questões primárias representam o ponto de partida para a revisão sistemática e são formuladas de maneira precisa e objetiva para orientar a seleção dos estudos a serem incluídos na revisão.

#### ***2.3.4.2 Questões secundárias***

Questões secundárias são perguntas específicas, mas complementares à pergunta principal, que podem abordar aspectos mais detalhados do problema em análise. As questões secundárias têm como objetivo orientar a busca, seleção, avaliação e análise da literatura relevante, bem como identificar lacunas no conhecimento e descrever as características dos estudos incluídos.

### **2.3.5 Palavras-chave**

As palavras-chave são termos que permitem aos pesquisadores identificar rapidamente estudos e artigos relevantes para a sua pesquisa. Essas palavras-chave podem ser usadas na busca de bancos de dados eletrônicos, como o PubMed, o Scopus e o Web of Science. Ao selecionar palavras-chave para uma revisão sistemática, é importante que os pesquisadores usem termos específicos que reflitam a questão de pesquisa.

### **2.3.5 Expressão de busca**

A expressão de busca é um conjunto de palavras-chave utilizadas para identificar artigos relevantes para a revisão sistemática, que podem estar contidos em diferentes bancos de dados. Essas palavras são selecionadas a partir de uma pergunta de pesquisa específica e de critérios de inclusão e exclusão previamente definidos.

### **2.3.6 Processo de seleção de estudos primários**

O processo de condução de uma revisão sistemática envolve várias etapas, incluindo triagem e seleção de estudos que serão incluídos na revisão final. O processo de seleção de estudos primários em uma revisão sistemática envolve uma abordagem rigorosa e padronizada que garante a confiabilidade e a validade dos resultados da revisão.

#### ***2.3.6.1 Critério de inclusão***

Os critérios de inclusão devem ser cuidadosamente escolhidos para garantir que apenas os estudos relevantes para a questão da pesquisa sejam incluídos na revisão.

#### ***2.3.6.2 Critério de exclusão***

Os critérios de exclusão também devem ser cuidadosamente escolhidos para garantir que estudos que não sejam de alta qualidade ou que não atendam aos objetivos da questão de pesquisa sejam excluídos.

### **2.3.7 Avaliação da qualidade**

A avaliação de texto completo envolve a leitura dos estudos para determinar sua qualidade e relevância para a questão de pesquisa. Os estudos são avaliados usando uma ferramenta padronizada que avalia a qualidade do desenho do estudo, metodologia e relatório.

### **2.3.8 Classificação dos dados**

Para sintetizar e analisar os dados extraídos de artigos incluídos nos diversos tipos de levantamentos sistemáticos da literatura, algumas estratégias gerais são comumente utilizadas. Em geral, os dados dos artigos são tabulados e/ou apresentados graficamente de forma a descrever de maneira global os artigos incluídos no estudo secundário.

Os estudos primários identificados podem ser descritos de acordo com algumas características como, por exemplo, ano de publicação, fonte encontrada, tipos de ferramentas

de estudo, definição de protocolos, método de pesquisa e aplicação, tipo de sistema, instrumento de avaliação de qualidade, análise de resultados, entre outros.

### 3. Metodologia

Nesta etapa será descrito o método utilizado para condução da pesquisa e obtenção dos resultados, se são confiáveis e verificáveis.

A metodologia utilizada nesse trabalho foi desenvolvida com base na Revisão Sistemática da Literatura (RSL), através de pesquisa de artigos científicos publicados na base de dados Scopus, IEEE e ACM e ligados à temática em questão.

Segundo Kitchenham (2007), uma Revisão Sistemática da Literatura (RSL) é um tipo de revisão que coleta vários estudos de pesquisa e os resume para responder a uma pergunta de pesquisa usando métodos bem definidos. Através desse modelo de revisão pode-se apresentar uma avaliação mais precisa e confiável a respeito de um tema, permitindo assim, a reprodutibilidade dos resultados.

Para a condução da coleta de dados, foi utilizado a ferramenta online Parsifal<sup>1</sup> desenvolvida para apoiar pesquisadores na realização de revisões sistemáticas de literatura no contexto da Engenharia de Software. Dessa forma, pesquisadores distribuídos geograficamente podem trabalhar juntos em um espaço de trabalho compartilhado, projetando o protocolo e conduzindo a pesquisa (FREITAS, 2020).

Além de fornecer uma maneira de documentar todo o processo, a ferramenta o ajudará durante a condução de uma revisão sistemática da literatura. Durante a fase de planejamento, Parsifal permite que pesquisadores cadastrem objetivos, PICOC<sup>2</sup>, questões de pesquisa, string de pesquisa, palavras-chave e sinônimos.

Adicionalmente, usuários podem ainda especificar as fontes de pesquisa e os critérios de inclusão e exclusão adotados na RSL. Os critérios de inclusão dos artigos selecionados foram: artigos sobre ferramentas de correção de trabalhos do curso introdutório de programação de computadores; artigos sobre ferramentas que facilitam o ensino aprendizagem na correção de trabalhos de programação de computadores; trabalhos que avaliam o impacto de diferentes ferramentas elencadas no estudo de ensino de programação de computadores; linguagem de programação para o processo de ensino-aprendizagem do curso de programação.

Os critérios de exclusão envolvem todos os artigos que fazem fuga ao tema proposto, embora alguns podem fazer parte do estudo caso alguma característica não o exclua diretamente do tema. Dessa forma, foram excluídos os artigos com os seguintes critérios:

---

<sup>1</sup> <https://parsif.al/about/>

<sup>2</sup> PICOC significa população, intervenção, comparação, resultados e contexto.

artigos muito antigos, publicados antes de 2010; trabalho não trata especificamente da problemática do estudo; estudo não científico; artigos que não tem o propósito do tema escolhido.

A Parsifal também fornecerá mecanismos para construir uma lista de verificação de avaliação de qualidade e formulários de extração de dados (FREITAS, 2020). Durante a fase de planejamento, o **Parsifal** permite o cadastro dos objetivos, PICOC, questões de pesquisa, string de pesquisa, palavras-chave e sinônimos, selecionando as fontes, os critérios de inclusão e exclusão (KITCHENHAM; CHARTERS, 2007).

Embora não identificando tais informações imediatamente, o Parsifal pode fornecer mecanismos para construir uma lista de verificação de avaliação de qualidade e formulários de extração de dados (KITCHENHAM, 2007). Nesse estudo, a qualidade dos artigos determinou-se através dos critérios de inclusão, atendendo com base nas seguintes regras: S (sim) para critérios bem explicitados no artigo, com pontuação 1.0; P (parcialmente) para critérios implícitos, somando-se 0,5; N (não) para artigos que não se adequam de forma alguma aos critérios de inclusão, somando-se 0,0.

Durante a fase de condução, o Parsifal fornece recursos para importar arquivos bibtex e selecionar os estudos, encontrar duplicatas entre todas as diferentes fontes, realizar a avaliação e qualidade e extrair dados dos trabalhos. Nesta RSL, adotaremos o protocolo PRISMA, que é amplamente aceito para avaliações de intervenções e de revisões sistemáticas publicadas (SOUZA; BATISTA; BARBOSA, 2014).

Para facilitar a implementação do protocolo PRISMA, é importante construir uma lista com o que não pode ficar de fora em cada uma das etapas da revisão sistemática, desde o título, resumo e introdução, orientando-se pelos objetivos, com os principais pontos dos métodos e resultados que fazem diferença na qualidade do estudo e fechando com a conclusão.

Para a pesquisa com base na revisão sistemática da literatura, as etapas são apresentadas da seguinte forma:

### **3.1 Processo de busca e objetivo:**

Essa etapa identifica as ferramentas de tecnologias e estratégias, com as pesquisas de artigos publicados entre 2010 a 2022, em bancos renomados como IEEE, ACM e Scopus.

### **3.2 String de busca**

Para a obtenção de resultados primários da RSL. Utilizada nos motores de busca em todos os repositórios de dados dos artigos pesquisados, através dos seguintes strings de busca: ((eval\*OR assessment OR grad\* OR feedback OR rating OR analysis) AND (auto\* OR mechanical) AND (program\* OR implementation OR development) AND (tool\* Or instrument)).

### **3.3 Condução**

Após a leitura dos resumos dos 108 artigos selecionados pelos descritores, foram selecionados 08 artigos para análise mais detalhada, pelo critério de inclusão (abordagem do tema proposto e publicação entre 2010 a 2022) e exclusão daqueles que fizeram fuga ao tema e anteriores ao período de publicação definido.

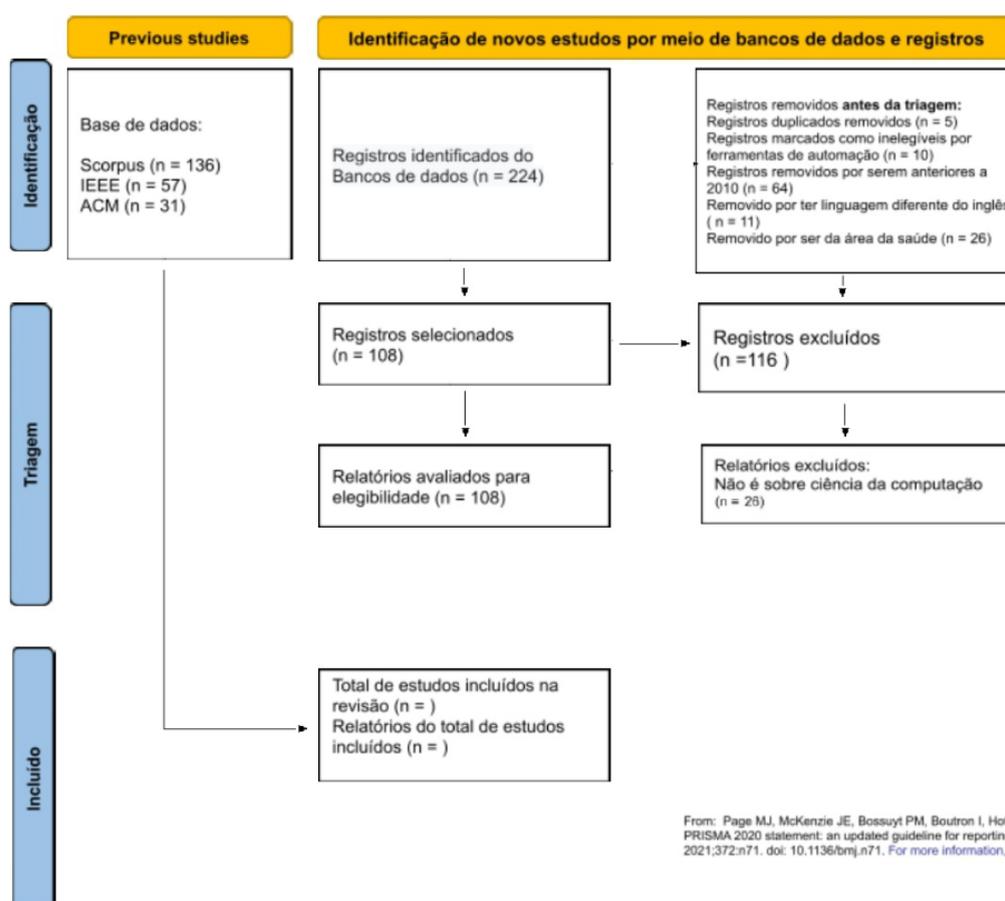
### **3.4 Análise de resultados**

Através dos artigos escolhidos por meio de um breve resumo, foram feitas leituras e análises mais aprofundadas dos resultados e análises dos ambientes e ferramentas de aprendizagem para o ensino e correção de trabalhos de cursos introdutórios de programação de computadores, mas apresenta resultados no próximo tópico.

## 4. Resultados

Foram encontrados 224 artigos a partir da busca dos descritores e dos strings de busca. Desse total, 136 foram encontrados na Scopus, 57 na IEEE e 31 na ACM. Antes da triagem foram removidos um total de 116 artigos, 5 deles por estarem duplicados; 10 por inelegibilidade; 64 artigos com data de publicação anterior a 2010; 11 artigos diferente da língua inglesa e 26 da área da saúde, restando 108 artigos para estudo, conforme demonstrado no Fluxograma 2.

**Fluxograma 3 - PRISMA 2020 para Revisões Sistemáticas**



Fonte: autora (2022).

### 4.1 Artigos escolhidos para estudo

Dos 108 artigos selecionados, apenas 8 foram selecionados e mantidos para os trabalhos apresentados, conforme Tabela 1, para este estudo. Os demais fizeram fuga ao tema proposto. Além disso, com o foco nas técnicas e ferramentas de correção de trabalhos e

de avaliação de código que podem ser usadas para fins educacionais, apenas as referências relevantes foram mantidas. Por exemplo, artigos focados em algoritmos, técnicas ou ferramentas para analisar código, mas não diretamente relacionados à avaliação automática de código para fins educacionais, foram deixados de lado.

**Tabela 1** – Relação dos artigos escolhidos para estudo

<b>Autor</b>	<b>Título</b>	<b>Ferramenta</b>
Barra et al., 2020	Avaliação automatizada em cursos de programação: um estudo de caso durante a era COVID-19	autoCOREctor
Tan et al., 2017	Codeflaws: uma competição de programação referência para avaliação de ferramentas automatizadas de reparo de programas	CODEFLAWS
Furia et al., 2015	AutoProof: verificação funcional autoativa de orientação a programas e objetos	AutoProof
Gütl et al., 2011	Criador automático aprimorado de perguntas - EAQC: Conceito, desenvolvimento e avaliação de um item de teste automático	EAQC
Rubio-Sanchez et al., 2013	Percepção do aluno e uso de uma ferramenta de avaliação de programação automatizada	Mooshak
Heckman e King, 2018	Desenvolvendo habilidades de engenharia de software usando ferramentas reais para classificação automatizada	Canary Framework
Heckman e King, 2018	Uma análise de ferramentas para desenvolvimento automático de software e geração automática de código	CASE e IDES
Gordillo 2019	Efeito de uma ferramenta centrada no instrutor para avaliação automática de atribuições de programação sobre as percepções e o desempenho dos alunos	Ferramenta centrada no instrutor

Fonte: própria autora (2023).

- **Avaliação automatizada em cursos de programação: um estudo de caso durante a era Covid-19** (BARRA et al., 2020). Este artigo foi publicado recentemente e compreende um pertinente estudo de caso feito durante a pandemia, utilizando a ferramenta autoCOREctor.
- **Codeflaws: uma competição de programação referência para avaliação de ferramentas automatizadas de reparo de programas** (TAN et al., 2017). Este artigo apresenta a ferramenta Codeflaws, que visa facilitar o estudo empírico futuro no reparo automatizado do programa.

- **AutoProof: verificação funcional autoativa de orientação a programas e objetos** (FURIA et al., 2015). Publicado na revista Nature Research Intelligence, este estudo se concentra na descrição da interface, design e recursos de implementação do AutoProof.
- **Criador automático aprimorado de perguntas - EAQC: Conceito, desenvolvimento e avaliação de um item de teste automático** (GÜTL et al., 2011). Neste artigo, que é uma versão estendida do artigo da conferência de Gütl, Lankmayr e Weinhofer (2010), apresenta-se o trabalho mais recente sobre a criação automatizada de diferentes tipos de itens de teste, mais precisamente, descreve-se o design e o desenvolvimento da ferramenta EAQC.
- **Percepção do aluno e uso de uma ferramenta de avaliação de programação automatizada** (RUBIO-SANCHEZ et al., 2013). Esse artigo descreve as principais características da ferramenta Mooshak, que funciona como um juiz automático para concursos de programação, além de avaliar a percepção do aluno.
- **Desenvolvendo habilidades de engenharia de software usando ferramentas reais para classificação automatizada** (HACKMAN e KING, 2018). Neste estudo, os autores apresentam a ferramenta Canary Framework para dar suporte às práticas de engenharia de software para profissionais em sala de aula.
- **Uma análise de ferramentas para desenvolvimento automático de software e geração automática de código** (HACKMAN e KING, 2018). Este outro estudo, dos mesmos autores, apresenta uma análise e avaliação de ferramentas para desenvolvimento automatizado de software e geração automática de código, a fim de determinar se elas atendem a um conjunto de métricas de qualidade.
- **Efeito de uma ferramenta centrada no instrutor para avaliação automática de atribuições de programação sobre as percepções e o desempenho dos alunos** (GORDILLO, 2019). Neste último artigo, da Universidade Politécnica de Madrid, na Espanha, o estudo examinou o efeito do uso de uma ferramenta centrada no instrutor para avaliar automaticamente tarefas de programação sobre as percepções e o desempenho dos alunos em um curso de desenvolvimento web em uma instituição de ensino superior.

## 4.2 Características das ferramentas de correção

Conforme demonstrado na Tabela 2, vários estudos estão focados nas ferramentas desenvolvidas para avaliação automatizada em cursos de programação.

Tabela 2 - Características das Ferramentas de Correção

Autores	Ferramenta	Avaliação Automatizada	Orientada para aprendizagem	Extraí conceitos aprendizagem textual	Juiz online	Centrada no aluno
---------	------------	------------------------	-----------------------------	---------------------------------------	-------------	-------------------

Barra et al., 2020	AutoCOREctor	Sim	Sim	Sim	Não	Sim
Tan et al., 2017	CODEFLAWS	Sim	Sim	Sim	Não	Não
Furia et al., 2015	EAQC	Sim	Sim	Sim	Não	Sim
Gütl et al., 2011	EAQC	Sim	Sim	Sim	Não	Sim
Rubio-Sanchez et al., 2013	Mooshak	Sim	Sim	Sim	Sim	Sim
Heckman e King, 2018	Canary Framework	Sim	Sim	Sim	Não	Sim
Heckman e King, 2018	CASE e IDES	Sim	Sim	Sim	Não	Não
Gordillo, 2019	Ferramenta Centrada no Instrutor	Sim.	Sim	Sim	Não	Não

Fonte: própria autora (2023)

Os estudos destacam principalmente, o grande número de diversas características possíveis de um lado e a dificuldade de desenvolvê-las de outro. No estudo de Barra et al. (2020), a ferramenta desenvolvida foi a AutoCOREctor, automatizada e orientada para aprendizagem centrada no aluno (somativa ou formativa). Também para avaliação e automatizada, a ferramenta Canary Framework, através do estudo de Heckman e King (2018).

A ferramenta CODEFLAWS, nos estudos de Tan et al. (2017), o reparo de programa automatizado de palavras-chave; classes de defeitos; avaliação empírica; referência são importantes características.

Já em Gütl et al. (2011), a ferramenta EAQC extrai conceitos mais importantes do conteúdo de aprendizagem textual, exercícios de conclusão e perguntas abertas. No estudo de Gordillo (2019), a ferramenta é centrada no instrutor e tem como característica avaliar automaticamente as tarefas de programação do aluno sobre as percepções e o desempenho dos alunos em curso de desenvolvimento web em uma instituição de ensino superior.

Segundo Rubio-Sanchez et al. (2013), a ferramenta Mooshak caracteriza-se como um juiz online que verifica a exatidão do programa, determinando sua utilidade em ambientes de sala de aula.

### 4.3 Feedback das ferramentas

Um fator fundamental das ferramentas de avaliação automática quando usadas para educação é a extração de uma medida comparando os programas submetidos a uma solução com relação aos objetivos de ensino. Para explicar essa medida para alunos e instrutores, o feedback resultante das avaliações é fundamental, como demonstrado na Tabela 3.

**Tabela 3** – Feedback das ferramentas

<b>Ferramenta</b>	<b>Feedback</b>	<b>Imediato</b>	<b>Mais elaborado</b>	<b>Menos elaborado</b>
autoCOREctor	Sim	Sim	Sim	não
AutoProof	Sim	Não	Não	Sim
EAQC	Sim	Sim	Sim	Não
Mooshak	Sim	Não	Não	Sim
Canary Framework	Sim	Sim	Sim	Não
CASE e IDES	Não	Não	Não	Não
Centrada ao instrutor	Sim	Não	Não	Sim

Fonte: própria autora (2023).

A ferramenta autoCOREctor apresentou um feedback imediato e mais elaborado (Barra et al. (2020). Já o AutoProof, de Furia et al. (2015), a verificação do feedback apresenta-se em duas etapas.

Enquanto EAQC fornece feedback, do estudo de Gütl et al. (2011), no estudo de Heckman e King (2018), as ferramentas CASE e IDES não apresentaram nenhum feedback. Nos estudos de Gordillo (2019), demonstrou que o feedback gerado pela ferramenta é difícil de entender, pouco útil com notas geradas injustas.

No outro estudo, também de Hackman e King, com a ferramenta Canary Framework, foi identificado um feedback rápido sobre erros de compilação, notificações de análise de programa e testes de unidade.

Ter sistemas capazes de produzir um bom feedback, especialmente os formativos, é um elemento chave para poder usá-los em ambientes educacionais. Segundo Heckman e King (2018), mais especificamente, um bom feedback deve explicar aos alunos como corrigir os problemas restantes e dar o próximo passo para suas tarefas.

Os sistemas desenvolvidos devem ter oportunidades para melhorar o nível de feedback fornecido aos alunos. Entretanto, ser capaz de automatizar a geração de feedback ajuda a tornar o processo de aprendizagem melhor, pois o feedback atua como um motivador constante (BARRA et al., 2020).

Ainda, de acordo com Barra et al. (2020), em alguns ambientes, o feedback instantâneo é necessário para que os alunos saibam imediatamente os erros que cometeram para melhorar a si mesmos e não repetir os erros novamente.

#### 4.4 Ferramentas e código fonte

Em relação aos métodos e técnicas usadas para avaliação automatizada de código, como demonstra a Tabela 4, alguns tipos de análises são predominantes nos artigos de revisão e pesquisa examinados.

**Tabela 4** – Ferramentas e código fonte

<b>Ferramenta</b>	<b>Código fonte</b>	<b>Aberto</b>	<b>Fechado</b>	<b>Integração</b>
AutoProof	Sim	Sim	Não	EVE
Mooshak	Sim	Sim	Não	
Canary Framework	Sim	Sim	Não	Jenkins
CASE e IDES	Sim	Sim	Não	XML

Fonte: própria autora (2023).

A ferramenta AutoProof, nos estudos de Furia et al. (2015), o código fonte é aberto e integrado ao EVE. Já na ferramenta Mooshak, dos estudos de Rubio-Sanchez et. al. (2013), o código fonte compila e executa para verificar se o programa gera as saídas desejadas, dadas as entradas iniciais. Outro conjunto de técnicas utilizadas conta com a presença de uma resposta correta de referência para as tarefas a serem trabalhadas pelos aprendizes.

Em um dos estudos de Heckman e King (2018), o código fonte (Jenkins) da ferramenta Canary Framework é aberto. Já no outro estudo, dos mesmos autores, os códigos das ferramentas CASE e IDES também são abertos, embora baseados em XML.

Segundo Gütl et al. (2011), ao trabalhar com avaliações baseadas em testes, é importante ter bons dados de teste, especialmente se eles forem gerados automaticamente. Também é importante determinar o nível em que as análises são realizadas.

Com relação ao uso específico na educação, esses sistemas podem ser vistos como mecanismos de aprendizado que melhoram a motivação, a progressão, as habilidades de autoavaliação e as habilidades relacionadas à ciência da computação dos alunos (TAN et al., 2017). O objetivo dessas plataformas é tentar medir se os recursos de aprendizagem dos instrutores foram alcançados.

Eles podem ser centrados em instrutores que precisam produzir notas ou em alunos que precisam praticar e melhorar suas habilidades. As ferramentas automatizadas de avaliação de

código podem ser usadas para avaliações formativas ou somativas em um ambiente totalmente ou semiautomático ou mesmo manual (GORDILLO, 2019).

Nesse mesmo contexto, de acordo com Furia et al. (2015), as ferramentas podem verificar uma variedade de programas que são representativos de idiomas orientados a objeto usados na prática.

No caso da avaliação formativa, segundo estudos de Barra et al. (2020), tais plataformas são uma oportunidade para fornecer conjuntos intensivos de exercícios para os alunos treinarem e passarem para a avaliação contínua de um curso.

Finalmente, muito pouca pesquisa e análise foram realizadas especificamente sobre a qualidade das avaliações produzidas. No entanto, todas as revisões examinadas concordam com o fato de que a avaliação e o feedback associado devem apoiar o processo de aprendizagem.

## **5. Considerações finais**

Esta pesquisa selecionou e analisou, através de uma Revisão Sistemática da Literatura, estudos sobre ferramentas para correção de trabalhos de cursos introdutórios de programação de computadores. Ficou demonstrado que a Revisão Sistemática de Literatura possui um método bem definido quanto à interpretação, identificação e avaliação de todas as pesquisas relevantes disponíveis em uma questão particular de pesquisa, sobretudo do tema em questão.

### **5.1 Conclusão**

Sabe-se que a disciplina de programação é fundamental em cursos de computação, além de exigir do aluno habilidades cognitivas como criatividade e raciocínio, bem como a resolução de problemas, através de diversas estruturas traduzidas em códigos.

Diante da importância do ensino de programação e das dificuldades em prover retorno às atividades propostas, faz-se necessária a utilização de ferramentas que realizem a correção automática de exercícios de programação. Dessa forma, o uso deste tipo de recurso permite que os alunos obtenham rapidamente um parecer sobre a correção de suas respostas, o que pode direcionar o estudo do conteúdo.

Em trabalhos relacionados, alguns teóricos avaliaram o impacto de diferentes ferramentas elencadas no estudo, e, constatou-se que a maioria foram satisfatórias para a correção e ensino de programação, destacando o Scrath e Alice como sendo os mais utilizados.

Pode-se perceber, através da pesquisa, sobre a importância que diferentes ferramentas apresentam para a correção de trabalhos e seu impacto no ensino de programação, bem como a as diversas propostas de ferramentas possuindo os mais diferentes recursos para a correção automática de exercícios de programação.

Foi percebido também nos estudos, as múltiplas linguagens de programação, bem como a possibilidade de definir casos de teste e a forma como as notas dos trabalhos são calculadas. Embora, em alguns casos, os aplicativos web enviados pelos alunos funcionaram corretamente em seus navegadores, mas produziram erros quando avaliados pela ferramenta de avaliação.

Com base nas evidências fornecidas por este estudo, pode-se sugerir que os sistemas de avaliação automatizados, quando adequadamente integrados aos cursos de programação, podem gerar uma melhoria significativa no desempenho dos alunos.

Para concluir, a revisão apresentada neste artigo abrange pesquisas recentes relacionadas a ferramentas para correção de trabalhos de programação e sistemas automatizados de avaliação de código.

Dessa forma, as questões de pesquisa abordadas neste artigo levaram a uma classificação das ferramentas recentes em diferentes eixos: as características das ferramentas, o código e os aspectos do programa que são avaliados, os métodos e técnicas utilizadas pelas ferramentas e os tipos de feedback gerado. A revisão também destacou várias maneiras de integrar ferramentas automatizadas de avaliação de código no processo de aprendizado.

## **5.2 Trabalhos e Atividades Futuros**

Diante das pesquisas, há evidência que os corretores automáticos de código, que auxiliam professor e aluno na correção de trabalhos de cursos introdutórios de programação de computadores são aplicadas e obtêm resultados satisfatórios, porém alguns estudos concluem que existe um espaço a ser melhor trabalhado em relação às estratégias e novas ferramentas tecnológicas para mediar o ensino da programação, deixando uma lacuna para trabalhos futuros.

Possíveis elementos de solução podem ser direções de trabalho interessantes para mais colaboração e para melhores sistemas modulares que podem ser facilmente usados para cobrir diferentes necessidades e ajudar os pesquisadores a trabalharem juntos.

## Referências

- ARIMOTO, M.; OLIVEIRA, W. **Dificuldades no Processo de Aprendizagem de Programação de Computadores: um Survey com Estudantes de Cursos da Área de Computação.** *In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI)*, 27, 2019, Belém. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2019, p. 244-254. ISSN 2595-6175. DOI: <https://doi.org/10.5753/wei.2019.6633>.
- BARRA, H., LOPEZ-PERNAS, S., ALONSO, A., SANCHEZ-RADA, J. F., GORDILLO, A., QUEMADA, J. **Avaliação automatizada em cursos de programação: um estudo de caso durante a era COVID-19.** *Dep. Eng. Sist. Telemáticos*. 2020.
- BEECROWD. **Comunidade global de desenvolvedores de softwares.** 2023. Disponível em: <https://beecrowd.io/en/home-page/> Acesso em: 03 fev. 2023.
- BERNARDI G.; CORDENONSI, A. Z. **O ensino de programação mediado por tecnologias educacionais: uma Revisão Sistemática de Literatura.** *Novas Tec. Educ.* 2020. v. 17, nº 1, p. 517-526.
- BRITO, P. S. S. **O uso de ferramentas computacionais para o ensino de programação para alunos de engenharia.** *Inst. Cienc. Ex. e Biol. UFOP*. 2019, 51 f.
- CHAVES, J. O. M; CASTRO, A. F; LIMA, R. W; LIMA, M. V. A; FERREIRA, K. H. A. **MOJO: Uma Ferramenta para Auxiliar o Professor em Disciplinas de Programação.** *In: CONGRESSO BRASILEIRO DE ENSINO SUPERIOR A DISTÂNCIA (ESUD 2013)*, 10, 2013, Belém.
- COSTA, R. R.; SOUSA, R. R. **O uso de tutores de programação inteligentes na produção de feedback para estudantes em tarefas de programação: uma revisão sistemática da literatura.** *Brazilian Journal of Development*. 2020. Doi: 10.34111/bjdv6n5-413.
- COSTA, A. B.; ZOLTOWSKI, A. P. C.; KOLLER, S. H.; TEIXEIRA, M. A. P. **Construção de uma escala para avaliar a qualidade metodológica de revisões sistemáticas.** *Rio de Janeiro: Cienc. Saúde colet.*, vol. 20, n. 8. 2015.
- FAPESP. **Alunos da USP criam sistema que corrige exercícios de programação em segundos.** 2015. Disponível em: <https://agencia.fapesp.br/alunos-da-usp-criam-sistema-que-corrige-exercicios-de-programacao-em-segundos/20890/> Acesso em: 20 jun. 2022.
- FERRI, J.; SELMA, S. S. R. **Como o ensino de programação de computadores pode contribuir com a construção de conhecimento na educação básica: uma Revisão Sistemática da Literatura.** *Novas Tec. Educ.* 2016. v. 14, n. 2, p. 1-10.
- FREITAS, V. S. **Parsifal.** 2020. Disponível em: <https://vitorfs.com/projects/parsifal/> Acesso em 20 jun. 2022.
- FUKUDA, D. Y. **Uma avaliação comparativa das linguagens Rust e C para ensino de programação.** *Inst. Cienc. Exatas*. 2021.

FURIA, C. A., NORDIO, M., POLIKARPOVA, N., TSCHANNEN, J. **AutoProof: verificação funcional autoativa de orientação a programas e objetos**. Univ. Tec. Gothenburg Sweden. 2015.

G1. **Mercado de TI sofre com a falta de profissionais**. 2020. Disponível em: <https://g1.globo.com/rj/sul-do-rio-costa-verde/especial-publicitario/unifoa/noticia/2020/10/26/mercado-de-ti-sofre-com-a-falta-de-profissionais.ghtml>. Acesso em: 08 maio 2022.

GERHARD, L. O.; BUSTAMANTE, I. M. R.; MAI, L. F. F.; NUNES, N. R.; SANDRINI, L. *et al* **Estudo de ferramentas de apoio à correção de atividades de programação no contexto do projeto IntroComp**. Cult. Dig. Educ. 2018, 2238-5916.

GORDILLO et al. **Efeito de uma ferramenta centrada no instrutor para avaliação automática de atribuições de programação sobre as percepções e o desempenho dos alunos**. 2019.

GUTL, C., LANKMAYR, K., WEINHOFER, J., HOFER, M. **Criador automático aprimorado de perguntas-EAQC**. Learn and Tech. Library. 2011.

HOLANDA, W. D.; FREIRE, L. P.; COUTINHO, J. C. S. **Estratégias de ensino-aprendizagem de programação introdutória superior: uma revisão sistemática da literatura**. Novas Tec. Educ. 2019., v. 17, n. 1, p. 527-536.

HECKMAN, S., KING, J. **Desenvolvendo habilidade de engenharia de software usando ferramentas reais para classificação automatizada**. Assoc. Comp. 2018, p. 794-799.

HECKMAN, S., KING, J. **Uma análise de ferramentas para desenvolvimento automático de software e geração automática de código**. Rev. Fac. Ing. Univ. Antioquia, 2018.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing Systematic Literature Reviews In Software Engineering**. 2007.

KITCHENHAM B. **Diretrizes para realizar revisões sistemáticas de literatura em engenharia de software**. Versão 2.3, relatório técnico EBSE EBSE-2007-01, Universidade de Keele e Universidade de Durham. 2007.

\_\_\_\_\_. **Procedures for performing systematic reviews Technical Report**. TR/SE-0401, Keele University and NICTA, 2004.

KUMAR S; GANKOTIYA, A. K; DUTTA, K. **A Comparative Study of Moodle with other e-Learning Systems**. In: INTERNATIONAL CONFERENCE ON ELECTRONICS COMPUTER TECHNOLOGY (ICECT 2011), 3, 2011, Kanyakumari, p. 414-418.

MONTEIRO, C. V. F. **Benefícios do uso de ferramentas de software nas fases iniciais do processo de inovação**. [S.l.], 2010

NICHOLSON, J.; MCCRILLIS, A.; WILLIAMS, J. D. **Collaboration challenges in systematic reviews: a survey of health sciences librarians**. J Med Libr Assoc. 2017, p. 385-93.

OLIVEIRA, S. O.; OLIVEIRA, A. A.; MOTA, L. C. **Uso de ambientes virtuais de aprendizagem como suporte ao ensino de programação: uma revisão sistemática.** Interf. Cient. Exatas e Tecnológicas. Aracaju. 2015, v. 1, n. 3, p. 9-22.

RAABE, A.; ZANCHETT, G.; VAHLDICK, A. Jogos de Programar como uma Abordagem para os Primeiros Contatos dos Estudantes com a Programação. Workshops do Congresso Brasileiro de Informática na Educação, 2015.

REIS, T.; MEDEIROS, H.; LOPES, R.; MEDEIROS, J. **Ensino-aprendizagem de programação: uma revisão sistemática da literatura.** Rev. Bras. Inform. Educ. 2015.

RICHTER, C. J.; BERNARDI, G.; CORDENONSI, A. Z. **O ensino de programação mediado por tecnologias educacionais: uma revisão sistemática de literatura.** [S.l.], 2019. v. 17, n. 1, 517–526 p.

RUBIO-SANCHEZ, M., FLORES, C. P., VELASQUEZ-ITURBIDE, J. A., KINNUNEN, P. **Percepção do aluno e uso de uma ferramenta de avaliação de programação automatizada.** Univ. Rey Juan Carlos, 2013.

SANTOS, F. A. O.; TELVINA, M. S.; SOUZA, H. S. **Code Teacher: ferramenta para correção automática de trabalhos de programação acadêmica em Java.** 2019. v. 5, n. 2, 1152 p. 1289-1301.

SILVA, D. N.; BRITO, J. R.; VAZ, N. A. P. **Lógica de programação: dificuldades de ensino-aprendizagem, métodos e ferramentas computacionais.** Cienc. Ex. e Tec. 2019.

SOUZA, D. M.; BATISTA, M. H. S.; BARBOSA, E. F. **Avaliação de qualidade de um ambiente de apoio ao ensino de programação.** Rev. Renote. 2014, v. 12, n. 2, p. 1-10.

SOUZA, D. M.; MALDONADO, J. C.; BARBOSA, E. F. **Aspectos de Desenvolvimento e Evolução de um Ambiente de Apoio ao Ensino de Programação e Teste de Software.** In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE 2012), 23, 2012, Rio de Janeiro.

SOUZA, F. H. **Crítérios Prisma para Revisões Sistemáticas.** 2015. Disponível em: <https://pt.slideshare.net/FabianoSouza22/prisma-revisoes-sistematicas-52046861>. Acesso em: 08 jun. 2022.

SPOJ BRASIL. **Sobre SPOJ Brasil.** 2023. Disponível em: <https://br.spoj.com/> Acesso em: 03 fev. 2023.

TAN, S. H., YI, J., YULIS., MECHTAEV, S., ROYCHOUDHURY, A. **Codeflaws: um benchmark de competição de programação para avaliar ferramentas automatizadas de reparo de programas.** 2017 IEEE/ACM 39ª Conferência Internacional sobre Software Engineering Companion (ICSE-C) , Buenos Aires, Argentina, 2017, pp. 180-182.

VIEIRA, E. C. et al. **Gestão do conhecimento na área de Tecnologia da informação: um estudo de caso da percepção sobre compartilhamento de conhecimento entre profissionais de**

TI. [S.l.],  
2021.

ZANETTI, H.; OLIVEIRA, C. **Práticas de ensino de Programação de Computadores com Robótica Pedagógica e aplicação de Pensamento Computacional**. n. Cbie, p. 1236, 2015.

ZHIGANG, S; XIAOHONG, S; NING, Z; YANYU, C. **Moodle Plugins for Highly Efficient Programmin Courses**. In: MOODLE RESEARCH CONFERENCE, 1, 2012, Heraklion, p. 157-163.

ZOLTOWSKI, A. P.; COSTA, A. B.; TEIXEIRA, M. A. P.,; KOLLER, S. H. **Qualidade metodológica das Revisões Sistemáticas em periódicos**. Teoria e Pesquisa. 2014, vol. 30, n. 1, p. 97-104.