

A survey of teaching methods for a programming subject: A literature review

ABSTRACT: Computer programming subjects make it possible to develop one of the main skills of students in computing courses. However, learning of programming is arduous, complex and requires a lot of dedication, both from professors and students. In this context, several approaches have been worked to improve the teaching and learning process of programming and, consequently, reduce the failure rate of students. Therefore, the purpose of this paper is to present a quasi-systematic literature review (RqSL), which was carried out aiming to discover in the specialized literature the approaches that are currently being used to support the teaching and learning process of programming in computing area. The results can contribute to future research by providing a set of approaches, as well as their good practices, results obtained in the implementation and difficulties of use identified in them.

Keywords: Algorithms, Programming, Teaching and Learning Process, Literature Review.

Um levantamento de métodos de ensino para a disciplina de programação: Uma revisão da literatura

RESUMO: As disciplinas de programação de computadores permitem desenvolver uma das principais competências dos alunos nos cursos na área da computação. No entanto, o aprendizado de programação é árduo, complexo e requer bastante dedicação, tanto dos docentes quanto dos discentes. Nesse contexto, diversas abordagens vêm sendo trabalhadas para melhorar o processo de ensino e aprendizado de programação e, conseqüentemente, reduzir a taxa de reprovação dos alunos. Sendo assim, a proposta deste trabalho é apresentar uma Revisão quasi-Sistemática da Literatura (RqSL), que foi realizada com o objetivo de descobrir na literatura especializada as abordagens que estão sendo utilizadas atualmente para apoiar o processo de ensino e aprendizagem de programação no ensino superior da área de computação. Os resultados podem contribuir para pesquisas futuras fornecendo um conjunto de abordagens, bem como suas boas práticas, resultados obtidos na implementação e dificuldades de utilização identificadas nas mesmas.

Palavras-chave: Algoritmos, Programação, Processo de Ensino e Aprendizagem, Revisão da Literatura.

Agradecimentos: Este trabalho pertence ao projeto SPIDER/UFPA (<http://www.spider.ufpa.br>).

1. INTRODUÇÃO

A programação de computadores é uma área de estudo essencial para a formação de discentes de diversos cursos na área da computação. Embora sua importância seja indiscutível, a complexidade de aprendizagem nas disciplinas de programação é elevada. Para GROTTA (2019, p.16) “aprender a programar requer uma alta carga cognitiva e uma longa curva de aprendizagem”.

GIRAFFA (2013) mostra que as disciplinas de programação são apontadas como principais responsáveis pelas grandes taxas de reprovação nos cursos de computação e contribuem de maneira significativa para a evasão dos alunos nos primeiros semestres dos cursos.

Nesse sentido, autores com VIEIRA *et al.* (2015) apresentam algumas das principais dificuldades dos alunos que podem estar relacionadas às grandes taxas de reprovação nas disciplinas de programação, como o pouco conhecimento em lógica de programação, dificuldades em identificar e abstrair características necessárias para o entendimento e a resolução dos problemas trabalhados.

O desenvolvimento da lógica de programação é comumente mais estimulado nas disciplinas introdutórias de programação, e isto requer uma atenção especial pois os alunos que não conseguem desenvolver totalmente as competências necessárias nessas disciplinas podem ter mais dificuldades nas demais disciplinas de programação de seus cursos. WATSON e LI (2014) mostraram em sua pesquisa que o índice de aprovação em disciplinas introdutórias de programação é de 67.7%.

Analisando o problema das altas taxas de reprovação das disciplinas de programação de computadores, torna-se de grande importância identificar e analisar a utilização diferentes métodos e técnicas de ensino e estudo que busquem potencializar a qualidade do ensino e aprendizagem em programação, para que assim seja possível elencar abordagens que estejam tendo êxito quando empregadas no ensino dessas disciplinas.

O presente trabalho buscou identificar na literatura especializada quais abordagens ou metodologias estão sendo empregadas atualmente no processo de ensino e aprendizagem de programação de computadores nos cursos relacionadas a área de computação, bem como os benefícios e desafios da aplicação das técnicas utilizadas. Para alcançar aos objetivos desta pesquisa, realizou-se uma Revisão quasi-Sistemática da Literatura (RqSL), que adotou boas práticas que se fazem presentes em uma Revisão Sistemática da Literatura (RSL), que para os autores KITCHENHAM e CHARTERS (2007), a RSL busca de forma técnica e criteriosa, identificar, coletar e avaliar dados que evidenciem fatos relevantes para a resposta de uma determinada questão de pesquisa.

Nesse sentido, a RqSL buscou identificar metodologias e/ou ferramentas que possam potencializar o ensino e aprendizagem de programação no nível superior. Os resultados da sua execução mostram diversos métodos que estão sendo utilizados atualmente no ensino de programação em cursos da área de computação do ensino superior.

Além desta seção introdutória, o presente artigo está estruturado da seguinte forma: a Seção 2 apresenta a fundamentação teórica; já a Seção 3 apresenta uma RqSL sobre trabalhos relacionados a essa pesquisa; a Seção 4 apresenta as etapas de execução da revisão, que consistem em planejamento, execução do método e resultados obtidos; na Seção 5 apresentamos as discussões relacionadas aos resultados da revisão; por fim, a seção 6 apresenta as conclusões com as limitações desse estudo e os trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

A seção de fundamentação teórica apresenta alguns conceitos para o melhor entendimento desse trabalho, tais como: ensino de programação.

2.1 Ensino de Programação

A programação de computadores é uma área que vem ganhando cada vez mais destaque na sociedade atual, que está cada vez mais envolvida com a utilização e o desenvolvimento de recursos tecnológicos. Isto faz com que o ensino de programação de computadores torne-se uma competência essencial para a geração atual, sendo classificada como uma das competências essenciais do século XXI (BERSSANETTE, 2018).

Diversos países como Estados Unidos e Estônia já adotam em seus currículos básicos o ensino de programação. No Brasil, o ensino de programação ocorre, principalmente, nos cursos de graduação e técnicos na área de computação. As Diretrizes Curriculares Nacionais (DCN's) para computação, reforçam que os cursos dessa área devem possuir uma base adequada de conteúdos que permitam o desenvolvimento de habilidades relacionadas ao raciocínio lógico e a programação de computadores (BRASIL, 2016).

Apensar das orientações presentes nas DCN's, não existe uma forma padronizada de ensinar programação, cada professor utiliza a abordagem que achar mais vantajosa para que se obtenha um bom rendimento de seus alunos. Dentre as abordagens utilizadas em salas de aula, podemos citar: a) o uso de gamificação, que consiste na aplicação de dinâmicas de jogos para engajar os alunos na resolução de problemas e conseqüentemente, potencializar o aprendizado (GONÇALVES *et al.*, 2019); b) o uso de mapas mentais, que são representações gráficas que auxiliam no encadeamento de ideias ou informações (LIU, TONG & YANG, 2018); c) *Problem Based Learning* (PBL) ou aprendizagem baseada em problemas, que é um método ativo que busca trabalhar na sala de aula a resolução de problemas que podem ser vivenciados no dia-a-dia de uma profissão (NOGUEIRA *et al.*, 2020); d) *coding dojo*, que é um método focado na aprendizagem ativa, onde os alunos aprendem um conteúdo de forma colaborativa aplicada à resolução de problemas (MARINHO *et al.*, 2016); entre outros.

Mesmo com tantas formas distintas de se ensinar programação, as disciplinas relacionadas à programação de computadores ainda possuem grandes taxas de reprovação. Uma pesquisa realizada na Universidade Federal Rural da Amazônia (UFRA) - Campus Capitão Poço (CCP) permitiu identificar o índice de reprovações de uma disciplina de programação de computadores nos cursos da área de computação do referido campus. Os dados mostraram que a taxa de reprovação na maioria das turmas analisadas por 6 semestres consecutivos, no período de 2017 até 2019, ultrapassam 30%, conforme ilustra a Figura 1.

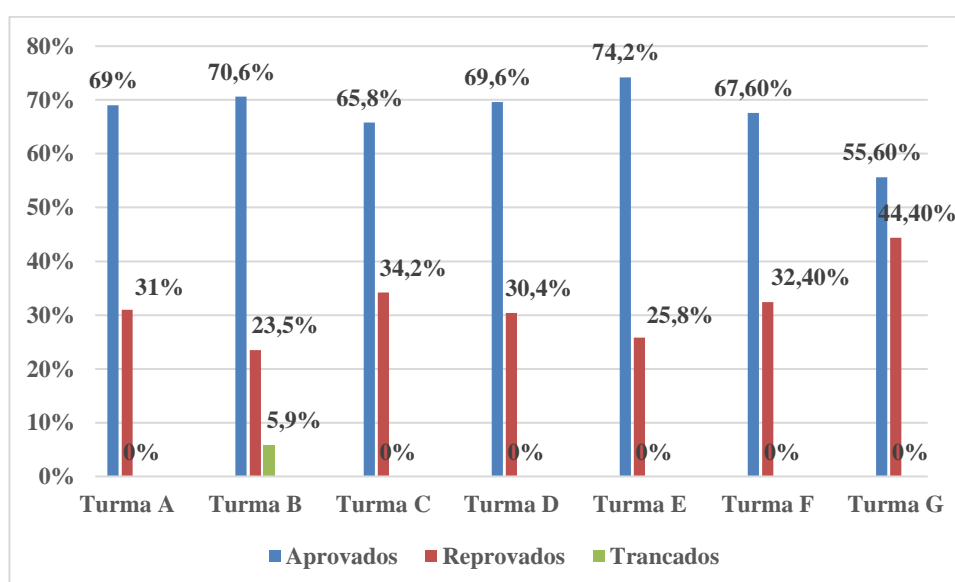


Figura 1 – Taxa de reprovação por turma analisada.

Fonte: Elaboração própria (2020).

Apesar da taxa de aprovados ainda ser superior que a de reprovação, a quantidade de alunos reprovados ainda é elevada, pois mesmo com diferentes professores lecionando a mesma disciplina a taxa mantém-se com um certo grau de similaridade, onde os aprovados oscilam entre 55,6% e 74,2% e os que não concluem a disciplina, seja por reprovação ou trancamento, ficam dentro de um intervalo que varia entre 25,8% e 44,4%.

Estas taxas de reprovação coincidem com pesquisas realizadas por RAMOS *et al.* (2015), que realizaram uma revisão da literatura em busca de fatores que contribuíssem para a aprovação ou reprovação dos alunos nas disciplinas de programação, os quais identificaram que as taxas de reprovação nessas disciplinas estão em torno de 45%. Um dos fatores que pode ajudar a explicar estes elevados índices pode ser o fato de que no Brasil o ensino de programação de computadores não está inserido ainda no currículo básico dos estudantes. O mesmo é realizado principalmente em cursos de nível superior, fazendo com que os alunos ingressem na programação com uma base de conhecimentos prévios deficitária.

LUXTON-REILLY (2019) aponta algumas razões que contribuem para o insucesso de alunos nas disciplinas de programação, como por exemplo: a) nível de conforto dos alunos com o conteúdo abordado; b) não receber ajuda suficiente da equipe do curso; c) dificuldade em entender o conteúdo; d) pouco tempo para dedicar aos estudos.

PARSONS *et al.* (2015) apontam em sua pesquisa que as taxas de reprovação em programação podem também estar relacionadas ao uso de métodos de avaliação inadequados, quem não conseguem avaliar as reais habilidades e competências que um verdadeiro programador deve possuir. Nesse sentido, torna-se de extrema importância investigar boas práticas que estão sendo adotadas atualmente nos cursos de programação, que busquem prover melhorias no ensino, de forma a engajar mais os alunos e elevar o índice de aprovação nas disciplinas.

3 TRABALHOS CORRELATOS

QUARESMA *et al.* (2019) apresentam os resultados de uma Revisão Sistemática da Literatura (RSL), a qual é focada na utilização de abordagens gamificadas no processo de ensino e aprendizagem de algoritmos. A RSL buscou por trabalhos dentro de um período de 10 anos, contemplando pesquisas publicadas entre 2006 a 2016. Isto permitiu selecionar e avaliar com base em critérios de inclusão e exclusão 61 trabalhos relacionados aos objetivos de sua pesquisa. Como resultados das análises dos trabalhos selecionados são apresentadas ferramentas, jogos computacionais, assim como metodologias que podem ser aplicadas no ensino de algoritmos com destaque para a gamificação.

SILVA *et al.* (2015) conduziram uma RSL sobre ensino e aprendizagem de programação para todos os níveis de ensino em um período de 5 anos, 2009 a 2013, que foi focada em analisar os resultados de publicações em 6 bases de dados que consistiram em quatro importantes eventos na área de Informática na Educação: (i) Simpósio Brasileiro de Informática na Educação; (ii) Workshop de Informática na Escola; (iii) Workshop de Educação em Computação; e (iv) Simpósio Brasileiro de Jogos e Entretenimento Digital; e duas grandes revistas nacionais: (v) Revista Brasileira de Informática na Educação; e (vi) Revista Novas Tecnologias na Educação. O estudo listou e agrupou diversas metodologias, técnicas e abordagens que são utilizadas no ensino em programação.

MEDEIROS, SILVA e ARANHA (2013) realizaram uma RSL que permitiu identificar produções voltados para a aplicação de jogos digitais no ensino de programação. A pesquisa foi baseada no período de 5 anos (2008- 2012) e utilizou como base de dados 3 eventos nacionais na área de informática na educação: (i) Simpósio Brasileiro de Informática na Educação (SBIE); (ii) Workshop de Informática na Escola (WIE); (iii) Workshop de Educação em Computação (WEI); e 3 revistas nacionais: (iv) Revista Brasileira de Informática na Educação (RBIE); (v) Revista Novas Tecnologias na Educação (RENOTE);

(vi) Revista Informática na educação: teoria e prática (RITA). Os resultados apresentados mostram que as pesquisas daquele período focavam na estimulação de ensinar programação no ensino médio e a execução das mesmas concentrava-se em três regiões brasileira Sul (29%), Sudeste (18%) e Nordeste (53%).

AURELIANO e TEDESCO (2012) apresentam os resultados de uma RSL que buscou identificar e analisar artigos relacionados ao processo de ensino-aprendizagem de programação para iniciantes no Brasil. Os autores abrangeram um período de 10 anos (2002-2011), utilizando como fontes de buscas dois eventos nacionais na área, que foram: (1) Simpósio Brasileiro de Informática na Educação (SBIE); (2) Workshop de Informática na Escola (WIE). Os autores identificaram um aumento na realização de pesquisas na área, principalmente nas regiões Sul e Sudeste do Brasil. A maioria das pesquisas tinham foco principal no ensino superior e boa parte delas apresentam propostas de novas ferramentas de softwares que podem ser utilizados no ensino de programação.

Diante do exposto, pode-se notar que as pesquisas estão focadas em sua maioria na utilização de uma abordagem específica, como por exemplo o uso da gamificação, assim como não existe um foco exclusivo em abordagens voltadas apenas ao ensino superior, o que faz com que essa pesquisa apresente um diferencial em relação as demais, pois a mesma busca levantar por meio de uma RqSL as abordagens que estão sendo trabalhadas atualmente no ensino superior de programação nos cursos da área da computação.

4 REVISÃO DA LITERATURA

Revisão quasi-Sistemática da Literatura foi o método escolhido para a realização desta pesquisa, pois é baseada nas boas práticas de uma Revisão Sistemática da Literatura (RSL), que permitem realizar um profundo levantamento de dados presentes na literatura a fim de identificar aspectos que possam responder a determinadas questões de pesquisa.

A revisão foi classificada como quase-Sistemática, pois diferente de uma RSL, esta pesquisa não realizou comparações entre publicações identificadas na busca, mas buscou preservar o mesmo formalismo apresentado no processo de condução de uma RSL. Nesse sentido, tanto a RqSL quanto a RSL são classificadas com estudos secundários, tendo em vista que seus resultados são gerados a partir da análise de estudos primários.

4.1. Planejamento da Revisão e Condução da Revisão

O planejamento da RqSL permitiu identificar as etapas necessárias para a realização da mesma, bem como todos os artefatos que seriam gerados em cada atividade. A definição das atividades foi baseada nas boas práticas presentes em uma RSL. Assim, conseguiu-se dividir a RqSL em 9 atividades, que foram representadas no fluxograma apresentado na Figura 2.

A Figura 2, ilustra todas as etapas da revisão bem como os artefatos gerados e utilizados em cada uma delas para que fosse possível alcançar os objetivos propostos nesta pesquisa. Nesse sentido, nas próximas subseções será apresentado o detalhamento das atividades presentes na RqSL.

4.1.1. Questões de Pesquisa

Como questão principal de pesquisa para a realização desta RqSL definiu-se:

- QP1: Quais abordagens estão sendo utilizadas atualmente no processo de ensino e aprendizagem de programação nos cursos de nível superior na área de computação?

A questão principal, foi o que norteou a execução da pesquisa. A partir da QP1 conseguiu-se estabelecer uma série de questões secundárias que deveriam ser respondidas a partir da extração de informações contidas nos artigos selecionados na RqSL. A seguir, iremos identificar as questões secundárias:

- QS1: Qual a abordagem mais utilizada no processo de ensino e aprendizagem de programação?
- QS2: Quais as principais dificuldades encontradas ao utilizar as abordagens identificadas nesta pesquisa no ensino de programação?
- QS3: Como se realiza o ensino de programação com as abordagens identificadas na pesquisa?
- QS4: Como foi feita a avaliação do processo de ensino e aprendizagem de programação com a utilização as abordagens identificadas na pesquisa?

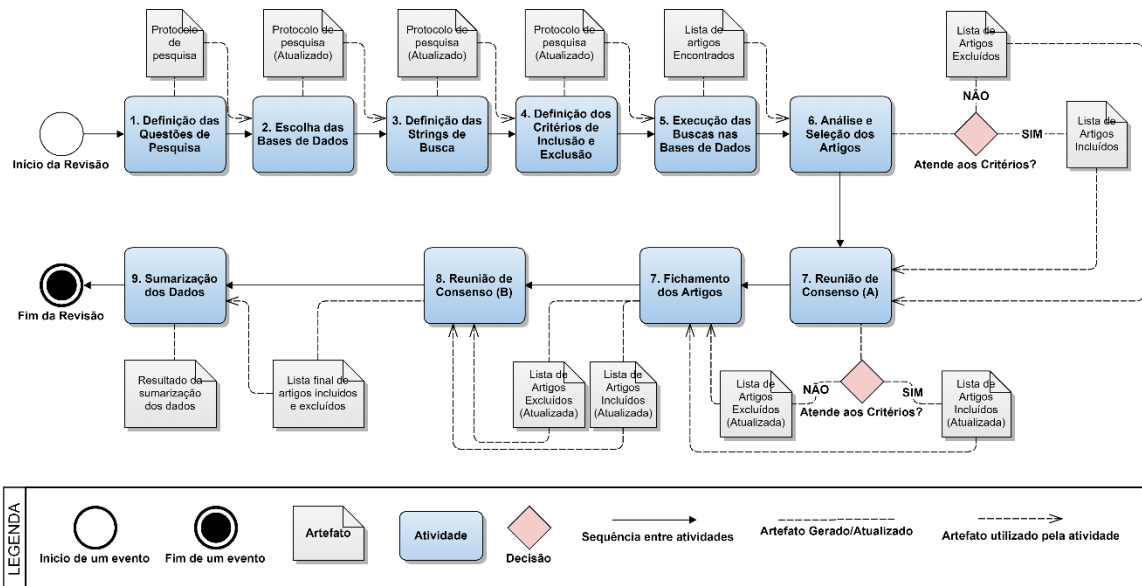


Figura 2 – Etapas da RqSL
 Fonte: Elaboração própria (2020).

4.1.2. Definição das bases de dados

Este trabalho realizou buscas automatizadas e manuais. Como fontes de buscas automatizadas adotou-se as bases de dados identificadas pelos IDs 1, 2, 3 e 4 ilustrados no Quadro 1, enquanto que as bases manuais estão representadas no referido quadro sendo identificadas pelos IDs 5 e 6.

Quadro 1 – Bases de dados utilizadas.

ID	Bases de dados utilizadas
1	IEEEExplore
2	Engineering Village (El Compendex)
3	Elsevier Scopus
4	ACM Digital Library
5	Congresso Brasileiro de Informática da Educação - CBIE
6	Workshop sobre Educação em Computação - WEI

Fonte: Elaboração própria (2020).

4.1.3. Definição dos critérios de busca

A *string* de busca utilizada neste trabalho foi gerada a partir de termos relacionados às palavras-chave, os quais foram baseados no critério PICO - *Participant* (Tipo de Participantes); *Intervention* (Tipo de Intervenção); *Comparasion* (Comparação); *Outcomes* (Tipo de resultados), que é um método bastante eficiente e prático para se construir questões

de pesquisas (NASCIMENTO *et al.*, 2018). Entretanto, para esta pesquisa utilizou-se apenas os itens *Population*, *Intervention* e *Outcomes* (PIO), pois para este trabalho não foi considerado relevante a realização de comparações. O Quadro 2 apresenta a estratégia PIO adaptada aos objetivos desta pesquisa.

Quadro 2 – Método PICO

Método PICO	
<i>Population</i> (Participantes)	Estudantes de programação do ensino superior brasileiro
<i>Intervention</i> (Intervenção)	Estratégias de ensino de programação associadas ao ensino de programação
<i>Outcomes</i> (Resultados)	1) Abordagens e métodos associados ao ensino de programação; 2) Principais benefícios proporcionados aos alunos; 3) Impactos do uso das estratégias no ensino e aprendizagem de programação.

Fonte: Elaboração própria (2020).

A partir da abordagem PIO conseguiu-se gerar a *string* de busca utilizada nesta pesquisa. Uma *string* de busca consiste no agrupamento de palavras-chave através dos operadores lógicos “OR” e “AND”. Nesse sentido, MALCHER *et al.*, (2015) destacam que deve-se utilizar o operador “OR” para agrupar palavras-chave e sinônimos, com base no método de pesquisa utilizado, enquanto que a utilização do operador “AND” é recomendada quando se deseja agrupar um conjunto de palavras-chaves que foram estabelecidos para todos os elementos. O Quadro 3 apresenta a *string* de busca gerada a partir da utilização dos operadores “OR” e “AND”, bem como o agrupamento das palavras-chaves e seus sinônimos.

Quadro 3 – *String* de busca.

String de busca
("programming learning" OR "programming teaching" OR "academic" OR "teaching" OR "learning" OR "undergraduation") AND ("process of teaching and learning in programming" OR "introduction to programming") AND ("model" OR "framework" OR "method" OR "technique" OR "methodology" OR "approach" OR "knowledge" OR "activity" OR "tool" OR "teaching tools")

Fonte: Elaboração própria (2020).

A revisão descrita pesquisou artigos publicados em periódicos científicos, conferências, simpósios e workshops durante o período de 01 de janeiro de 2016 até 31 de dezembro de 2019, contemplando, desta forma, um período de 04 anos. O objetivo de trabalhar com os últimos 4 anos foi identificar apenas abordagens que estão sendo utilizadas atualmente no ensino de programação.

A *string* de busca foi testada, adaptada e avaliada a partir de diversos testes nas fontes de busca, as validações dos testes foram baseadas nos trabalhos de QUARESMA *et al.* (2019), SPRINT e COOK (2015) e HIRJON-NEIRA *et al.* (2014).

4.1.4. Critérios de Seleção dos Estudos Primários e Processo de Seleção

Com a finalização do processo de busca nas bases de dados, iniciou-se o processo de seleção dos trabalhos relevantes, os quais foram norteados por duas etapas: 1ª etapa, dois pesquisadores realizaram a análise dos trabalhos com base na leitura dos títulos, resumos e

palavras-chave; 2ª etapa, após cada etapa de seleção individual, os pesquisadores realizaram reunião de consenso, na ocorrência de divergências entre os pesquisadores, isto é, quando não houve uma unanimidade na inclusão de um artigo e pelo menos um pesquisador decidiu incluir, o artigo era automaticamente selecionado. Os pesquisadores basearam-se nos critérios de inclusão e exclusão apresentados no Quadro 4.

Quadro 4 – Critérios de seleção dos estudos primários

ID	Critério de Inclusão (CI) ou Critério de Exclusão (CE)
CI 1	Publicações de estudos primários que apresentem o uso de metodologias educacionais no processo de ensino e aprendizagem de programação no ensino superior na área de computação.
CE 1	Publicações que não tenham foco em cursos do ensino superior.
CE 2	Publicações que não estejam disponíveis na íntegra através do domínio da Universidade Federal do Pará (UFPA) ou a partir da utilização da <i>engine</i> de busca Google e/ou Google Scholar.
CE 3	Publicações do tipo relatórios de workshops, apresentações, pôster, <i>keynotes</i> e <i>speaker</i> .
CE 4	Artigos publicados fora do período definido para a busca.
CE 5	Artigos que não estejam nos idiomas Inglês ou Português.
CE 6	Artigos que claramente não atendam as questões de pesquisa.
CE 7	Artigos duplicados, neste caso apenas a primeira ocorrência será considerada.

Fonte: Elaboração própria (2020).

Após a seleção dos estudos que obedeceu aos critérios estabelecidos no Quadro 4, realizou-se a sumarização dos dados extraídos dos artigos. Os resultados da extração dos dados serão apresentados na Seção 5 deste trabalho.

5 RESULTADOS E DISCUSSÕES

Nesta seção apresentam-se os resultados e as discussões como forma de resposta às questões de pesquisa.

5.1. Resultados da RsQL

Após a execução da *string* de busca nas bases de dados automáticas foram retornados 785 trabalhos, sendo 251 pertencentes a base IEEEExplore, 30 da Engineering Village (El Compendex), 314 da base de dados Elsevier Scopus e 190 da ACM Digital Library. Realizou-se também buscas manuais nas bases de dados do CBIE com 172 artigos retornados, enquanto que o WEI retornou 57 resultados, totalizando 229 trabalhos. A somatória dos artigos localizados nas bases de dados automáticas e manuais foi igual a 1014 artigos retornados, conforme mostra a Tabela 1.

Tabela 1 – Artigos retornados por base de dados

ID	Bases de dados utilizadas	Quantidade de artigos retornados
1	IEEEExplore	251
2	Engineering Village (El Compendex)	30
3	Elsevier Scopus	314
4	ACM Digital Library	190
5	Congresso Brasileiro de Informática da Educação - CBIE	172
6	Workshop sobre Educação em Computação –	57

ID	Bases de dados utilizadas	Quantidade de artigos retornados
	WEI	
	TOTAL	1014

Fonte: Elaboração própria (2020).

Após o download dos trabalhos encontrados, realizou-se uma busca com o objetivo de identificar trabalhos duplicados em diferentes bases de dados, os quais foram removidos, permanecendo apenas a primeira ocorrência do trabalho. Posteriormente, os trabalhos restantes foram analisados com base em seus títulos, resumos e *abstracts*, onde os critérios de inclusão e exclusão foram aplicados nos referidos itens. Desta forma, foi possível identificar 50 artigos que se encaixavam com os objetivos desta pesquisa, os quais foram selecionados para análise, e em seguida os resultados foram discutidos através de uma reunião de consenso, que permitiu chegar aos resultados ilustrados na Tabela 2.

Tabela 2 – Quantidade de artigos baixados, duplicados, incluídos e excluídos

Bases de dados utilizadas	Artigos Retornados nas Buscas	Artigos Baixados	Artigos depois de removidas as duplicatas	Artigos incluídos	Artigos Excluídos
IEEEExplore	251	246	242	6	245
Engineering Village (El Compendex)	30	19	2	0	30
Elsevier Scopus	314	223	201	11	303
ACM Digital Library	190	184	159	12	178
Congresso Brasileiro de Informática da Educação - CBIE	172	172	172	8	164
Workshop sobre Educação em Computação – WEI	57	57	54	13	44
TOTAL	1014	901	830	50	964

Fonte: Elaboração própria (2020).

Com base na Tabela 2, pode-se notar que dos 1014 artigos retornados nas buscas foi possível ter acesso a 901, pois 150 foram excluídos com base no CE 2. Em seguida, buscou-se identificar artigos duplicados, onde foram encontradas 71 ocorrências de trabalhos repetidos, os quais foram eliminados com base no CE 7. Após a remoção das duplicatas, foram realizadas as seleções do trabalho com base na leitura dos mesmos. O total de exclusões considerando os critérios estabelecidos foi de 964 artigos. A Tabela 3 apresenta as estatísticas de aplicação dos critérios de Inclusão e de Exclusão.

Tabela 3 – Estatísticas de Aplicação dos Critérios de Inclusão e Exclusão

Base de dados	Quantidade por Critério							
	CI 1	CE 1	CE 2	CE 3	CE 4	CE 5	CE 6	CE 7
IEEEExplore	6	7	5	3	0	2	224	4
Engineering Village (El Compendex)	0	0	11	0	0	0	2	17
Elsevier Scopus	11	10	91	4	0	3	173	28

Base de dados	Quantidade por Critério							
	CI 1	CE 1	CE 2	CE 3	CE 4	CE 5	CE 6	CE 7
ACM Digital Library	12	39	6	5	0	0	103	25
Congresso Brasileiro de Informática da Educação - CBIE	8	54	0	0	0	0	110	0
Workshop sobre Educação em Computação – WEI	13	19	0	0	0	0	22	3
Total	50	129	113	12	0	5	634	77

Fonte: Elaboração própria (2020).

Como pôde-se observar anteriormente na Tabela 3, houve uma grande quantidade de ocorrências dos critérios CE 1, CE 2 e CE.6. No critério CE 1, onde foram excluídas todas as publicações cujo foco não fossem o ensino superior, analisou-se os artigos disponíveis e identificou-se que 129 foram eliminados com base no referido critério. Com relação ao critério CE 2, verificou-se que como a base de dados da Scopus foi a última base a ser avaliada entre as automáticas, a mesma apresentou a maior quantidade de trabalhos duplicados, pois os mesmos já haviam sido identificados nas demais bases automatizadas. O critério CE 6 foi o mais aplicado na filtragem de artigos que não correspondiam ao escopo da pesquisa, por não mencionarem as palavras-chaves de busca em seu título ou resumo, permitindo eliminar 634 trabalhos. O critério de inclusão CI 1 foi o que direcionou a seleção de trabalhos que estivessem aderentes ao escopo da pesquisa, permitindo a identificação e seleção de 50 trabalhos.

Outra restrição era o critério CE 4, referente ao período temporal da busca de materiais, que abrangeu o período de 01 de janeiro de 2016 à 31 de dezembro de 2019. Nesse sentido, pode-se observar que nenhum trabalho foi excluído com base neste critério.

Os trabalhos selecionados foram classificados por ano, o que permitiu a elaboração do gráfico apresentado na Figura 3, onde pode-se notar que o ano de 2016 apresenta 18 trabalhos e no ano de 2019 foram 17, sendo os anos que obtiveram uma quantidade maior de artigos selecionados relacionados ao ensino de programação de computadores, enquanto que o ano 2017 foi o que apresentou menos resultados.

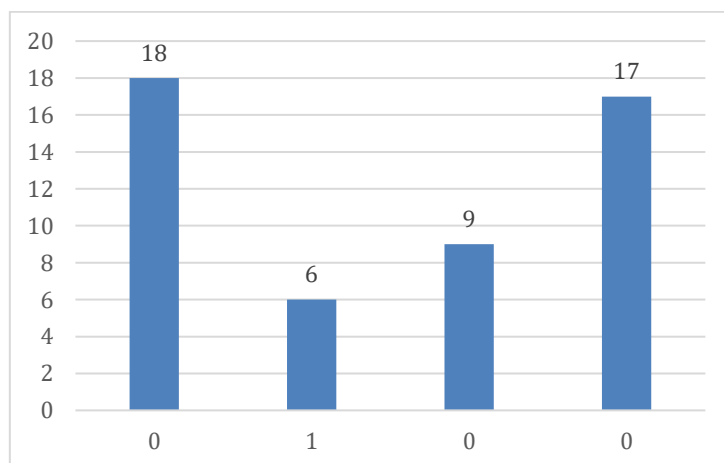


Figura 3 – Classificação dos trabalhos por ano de publicação.

Fonte: Elaboração própria (2020).

Também buscou-se identificar a origem das publicações, a fim de conhecer os países que estão trabalhando com as metodologias que busquem a melhoria do processo de ensino e aprendizagem de programação, conforme ilustra o gráfico apresentado na Figura 4.

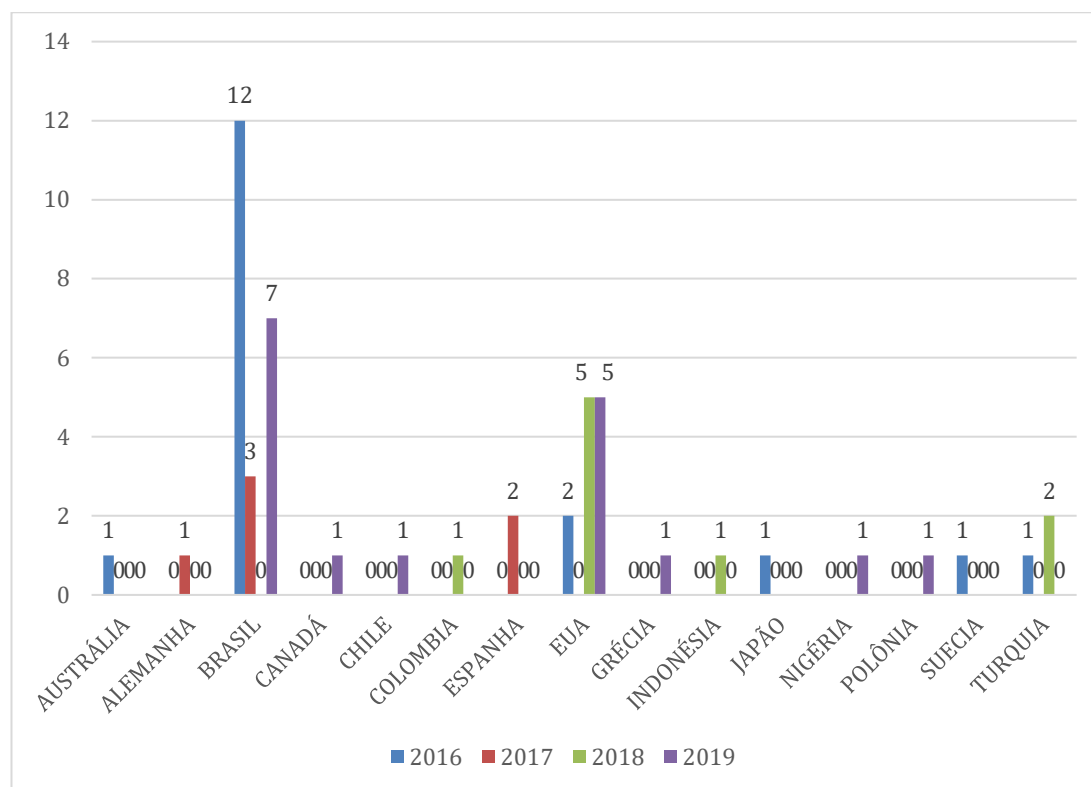


Figura 4 – Classificação dos trabalhos por país e ano de publicação.

Fonte: Elaboração própria (2020).

Ao analisar os dados apresentados anteriormente na Figura 4, pode perceber que o Brasil foi o país que mais teve o maior número de publicações, com 12 no ano de 2016, 3 em 2017 e 7 em 2019. Em segundo lugar os Estados Unidos da América (EUA) destacam-se com 4 publicações em 2019, 5 no ano de 2018 e 2 em 2016. Um dos fatores que podem explicar o Brasil ter mais publicações pode ser o fato de que a base de dados de 2 grandes eventos nacionais na área pesquisa foram utilizadas como fontes de busca.

Realizou-se também a classificação por autores, levando em conta a ocorrência do nome do primeiro autor dos trabalhos selecionados. A Tabela 4 detalha a relação completa dos autores identificados nos artigos, onde pode-se notar que os autores “Özyurt, Hacer” e “Raj, Adalbert Gerald Soosai” destacam-se com a maior quantidade de publicações referentes aos objetivos desta pesquisa, que busca por métodos utilizados no ensino superior dos cursos de computação que objetivem a melhoria do processo de ensino e aprendizagem de programação.

Tabela 4 – Classificação de autores por quantidade de publicação

Autor	Quantidade	Título
Özyurt, Hacer	2	Analyzing the effects of adapted flipped classroom approach on computer programming success, attitude toward programming, and programming self-efficacy
		Using Facebook to enhance learning experiences of students in computer programming at Introduction to Programming and Algorithm course

Autor	Quantidade	Título
Raj, Adalbert Gerald Soosai	2	Is More Active Always Better for Teaching Introductory Programming? Role of live-coding in learning introductory programming
Alves, Géssica	1	Coding Dojo como Prática de Aprendizagem Colaborativa para Apoiar o Ensino Introdutório de Programação: Um Estudo de Caso
Anfurrutia, Felipe I.	1	Visual Programming Environments for Object-Oriented Programming: Acceptance and Effects on Student Motivation
Brauner, Daniela	1	Estímulo à prática multidisciplinar no ensino de Computação e Design através de um evento de programação focado em problemas
Brown, Michael Scott	1	Dynamic hierarchical learning material for educational institutions
Cain, Andrew	1	Reflections on applying constructive alignment with formative feedback for teaching introductory programming and software architecture
Carvalho, Leandro S G	1	Ensino de Programação para Futuros Não-Programadores : Contextualizando os Exercícios com as Demais Disciplinas de mesmo Período Letivo
Cristina, Viviane	1	Desafios e oportunidades aos processos de ensino e de aprendizagem de programação para iniciantes
De Oliveira, Silmar Antonio Buchner	1	Programação para Administração de Redes de Computadores - Uma Experiência com Estudantes de Computação
Färnqvist, Tommy	1	Supporting active learning by introducing an interactive teaching tool in a data structures and algorithms course
Galvão, Leandro	1	Juiz online como ferramenta de apoio a uma metodologia de ensino híbrido em programação
Garces, Sebastian	1	Effects of Self-explanations as Scaffolding Tool for Learning Computer Programming
Gonçalves, Benediel	1	Elementos de Gamificação Aplicados no Ensino-Aprendizagem de Programação Web
Kentros, Sotirios	1	Course Redesign to Improve Retention : Finding the Optimal Mix of
Krugel, Johannes	1	Computational thinking as springboard for learning object-oriented programming in an interactive MOOC
Kutzke, Alexander Robert	1	Mediação do erro no ensino de programação de computadores: fundamentos e aplicação da ferramenta FARMA-ALG
Lima, Árlon	1	Metodologia 7Cs: Uma Nova Proposta de Aprendizagem para a Disciplina Algoritmos
Marín, B.	1	An empirical investigation on the benefits of gamification in programming courses
Marinho, Carlos	1	Experiências no Uso da Metodologia Coding Dojo nas Disciplinas Básicas de Programação de Computadores em um Curso Interdisciplinar do Ensino Superior
Martínéz- Valdés, José Alfrédo	1	A (relatively) unsatisfactory experience of use of Scratch in CS1
Matsuzawa, Yoshiaki	1	Measuring an impact of block-based language in introductory programming

Autor	Quantidade	Título
McDonald, Daniel	1	Teaching C# Using Xamarin and Android Tablets
Meira, Matheus C	1	Torneios Baseados em Robocode para Incentivar Jovens a Aprender Programação
Mohamed, Abdallah	1	Designing a CS1 programming course for a mixed-ability class
Montagner, Igor S	1	Teaching C programming in context: a joint effort between the Computer Systems, Embedded Computing and Programming Challenges courses
Mutiawani, Viska	1	Markas C: An E-Learning Media for Introductory Programming
Netto, Dorgival	1	Game Logic : Um jogo para auxiliar na aprendizagem de lógica de programação .
Norouzi, Narges	1	Quantitative evaluation of student engagement in a large-scale introduction to programming course using a cloud-based automatic grading system
Nunes, Ricardo Rodrigues	1	SimProgramming: uma abordagem motivacional para a aprendizagem de alunos intermediários de programação
Oliveira, Luiz	1	Desenvolvimento de um aplicativo móvel educacional voltado ao ensino de Estrutura de dados
Perenc, Izabela	1	Teaching programming using dedicated Arduino Educational Board
Phan, Vinhthuy	1	Code4Brownies: An active learning solution for teaching programming and problem solving in the classroom
Porter, Leo	1	A multi-institutional study of peer instruction in introductory computing
Quaresma, José	1	Uma Avaliação Experimental do Uso de um Framework Gamificado para a Disciplina Algoritmos e Equivalente
Rabêlo Júnior, Dilson José Lins	1	Cosmo: Um ambiente virtual de aprendizado com foco no Ensino de Algoritmos
Raposo, Ewerton Henning Souto	1	O Desafio da Serpente - Usando gamification para motivar alunos em uma disciplina introdutória de programação
Sakpere, Aderonke Busayo	1	Using social platforms to increase engagement in teaching computer programming
Sales, André Barros de	1	Utilização de Problemas da Maratona de Competição de Programação e Juízes Eletrônicos como Estratégia de Ensino em um Curso de Graduação em Engenharia de Software.
Santana, Bianca L	1	Motivação de Estudantes Non-Majors em uma Disciplina de Programação
Santiago, Almir David	1	Um modelo lúdico para o ensino de conceitos de programação de computadores
Schneider, Susan C.	1	The Impact of Adding an Interactive Textbook on Student Performance and Satisfaction in a Freshman Introduction to Programming Course
Sideris, Grigorios	1	PY-RATE ADVENTURES: A 2D Platform Serious Game for Learning the Basic Concepts of Programming With Python
Silva, Leuson M P	1	POOGame : Um Jogo Sério para o Ensino de Programação Orientada a Objetos

Autor	Quantidade	Título
Siva, Sebastien	1	Using music to engage students in an introductory undergraduate programming course for non-Majors
Tarimo, William	1	Early detection of at-risk students in CS1 using teachback/spinoza
Toledo, Javier Alejandro Jiménez	1	User interface sketch for the development of algorithmic thinking with augmented reality
Topalli, Damla	1	Improving programming skills in engineering education through problem-based game projects with Scratch

Fonte: Elaboração própria (2020)

Buscou-se também identificar as instituições que mais produziam pesquisas na área. Para levantar estas informações considerou-se apenas a instituição do primeiro autor dos artigos. A Figura 5 apresenta o resultado da quantidade de trabalhos por instituição, onde pode-se notar que a Universidade Federal do Pará ficou em primeiro lugar com 3 publicações, em segundo lugar foram identificadas quatro universidades: a) *University of Wisconsin-Madison*, b) *University of California*, c) Universidade Federal do Amazonas e d) Universidade Estadual de Campinas, ambas com 2 publicações.

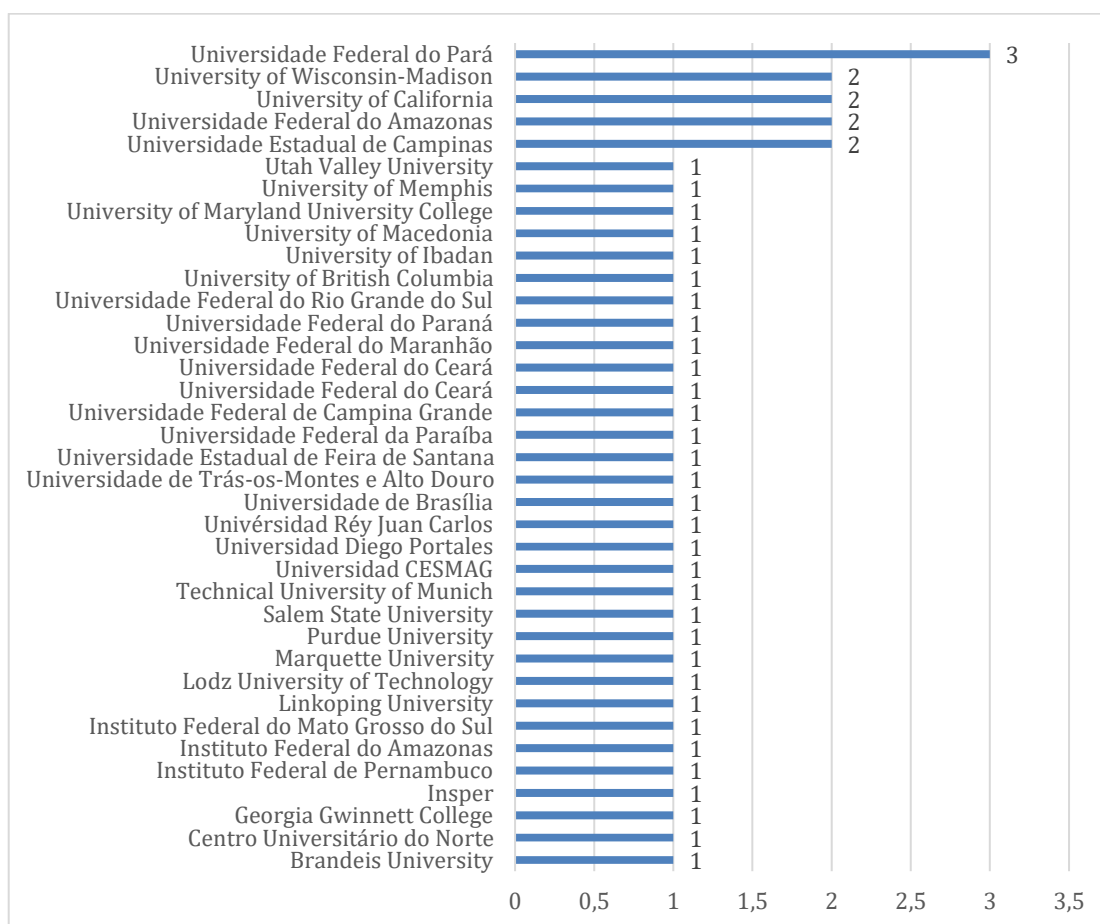


Figura 5 – Classificação dos trabalhos por país e ano de publicação.

Fonte: Elaboração própria (2020).

5.2 Análise das Questões de Pesquisa

As questões de pesquisa foram fundamentais para nortear a correta execução da RsQL, os resultados das análises dos artigos selecionados permitiram responder com sucesso a todas as questões de pesquisa presentes neste trabalho, sendo uma considerada a questão principal que foi a propulsora da execução desta pesquisa e quatro secundárias que permitiram explorar com mais detalhes a temática investigada na questão principal.

5.2.1 QP1: Quais abordagens estão sendo utilizadas atualmente no processo de ensino e aprendizagem de programação nos cursos de nível superior na área de computação?

A questão principal deste trabalho buscava identificar as abordagens que estão sendo utilizadas atualmente nos cursos superiores na área de computação que busquem a melhoria do processo de ensino e aprendizagem de programação. As análises dos artigos selecionados permitiram identificar 15 abordagens, conforme ilustra do Quadro 5. Vale ressaltar que alguns artigos utilizam mais de uma abordagem em sua proposta.

Quadro 5 – Abordagem trabalhada nos artigos.

ID	Abordagem
1	Serious Games
2	Propostas ou uso de Ferramentas
3	Coding Dojo
4	Sala de Aula Invertida
5	Gamificação
6	Ensino Híbrido
7	Desafios de Programação
8	Método Focado em Exercícios
9	Dispositivos Móveis
10	Redes Sociais
11	Ambiente Virtual
12	Programação em Tempo Real
13	<i>Problem Based Learning - PBL</i>
14	<i>Massive Open Online Course - MOOC</i>
15	Arduino

Fonte: Elaboração própria (2020).

5.2.2 QS1: Qual a abordagem mais utilizada no processo de ensino e aprendizagem de programação?

Para responder a QS1 buscou-se identificar dentre as abordagens encontradas a que teve maior ocorrência dentre os trabalhos analisados. Assim, pode-se notar com a Figura 6 que abordagem *Serious Games* aparece em primeiro lugar sendo utilizada em 12 trabalhos. É importante ressaltar que para os artigos que utilizaram mais de uma abordagem em sua proposta todas as abordagens foram contabilizadas, cada qual em sua categoria correspondente.



Figura 6 – Classificação dos trabalhos por país e ano de publicação.

Fonte: Elaboração própria (2020).

5.2.3 QS2: Quais as principais dificuldades encontradas ao utilizar as abordagens identificadas nesta pesquisa no ensino de programação?

A QS2 busca investigar as dificuldades ou pontos fracos da utilização das abordagens analisadas neste trabalho. Assim, conseguiu-se responder a esta questão secundária com alguns relatos dos trabalhos analisados, conforme o Quadro 6, que foi elaborado com base nas dificuldades relatadas nos artigos que utilizam uma determinada abordagem.

Quadro 6 – Dificuldades identificadas na utilização das abordagens

Abordagem	Dificuldades na utilização
Serious Games	Fatores como falta de incentivo por parte da instituição, falta de instruções sobre o uso correto da abordagem e descrédito dos alunos em relação ao uso de Jogos Sérios tendem a dificultar a utilização da abordagem.
Propostas ou uso de Ferramentas	Dificuldades no manuseio e no entendimento das ferramentas tendem a prejudicar a utilização das mesmas.
Coding Dojo	A falta de experiência em trabalhos em duplas ou em conjunto pode ser um fator que dificulte a utilização, bem como a diferença de conhecimento entre os alunos.
Sala de Aula Invertida	O uso da técnica de sala de aula invertida requer um trabalho adicional de professores e alunos para que se crie um hábito disciplinado de estudar o conteúdo extraclasse.
Gamificação	
Ensino Híbrido	Uma das dificuldades apresentadas está relacionada ao estudo dos alunos online bem como sanar as dúvidas dos mesmos remotamente, pois nem sempre as respostas que os alunos precisam, estão na internet.
Desafios de Programação	Normalmente essa abordagem é utilizada na forma de eventos o que tende a dificultar seu emprego dentro das disciplinas de programação.
Método Focado em Exercícios	Não se realizou avaliação prévia a fim de medir o conhecimento e o entendimento dos participantes em relação a abordagem utilizada.

Abordagem	Dificuldades na utilização
Dispositivos Móveis	A não disponibilidade de dispositivos móveis para todos os alunos, fazendo com que os mesmos tenham que utilizar seus próprios smartphones.
Redes Sociais	Não se identificou nos artigos relatos de dificuldade de utilização.
Ambiente Virtual	Percebeu-se decréscimo no rendimento dos alunos a medida em que a dificuldade das tarefas iam aumentando.
Programação em Tempo Real	Não se tem registros de utilização com turmas de programação avançada, apenas com turmas iniciantes, essa ausência de informações pode dificultar implementações da abordagem em um público alvo de programação avançada.
<i>Problem Based Learning</i> – PBL	Não se identificou nos artigos relatos de dificuldade de utilização.
<i>Massive Open Online Course</i> – MOOC	Não se identificou nos artigos relatos de dificuldade de utilização.
Arduino	Não se identificou nos artigos relatos de dificuldade de utilização.

Fonte: Elaboração própria (2020).

5.2.4 QS3: Como se realiza o ensino de programação com as abordagens identificadas na pesquisa?

A QS3 foi elaborada com o objetivo de identificar como está sendo realizado o ensino de programação com base nas abordagens analisadas neste trabalho. Para responder a QS3 foram extraídas dos artigos selecionados as metodologias de aplicação da abordagem adotada. Os resultados permitiram elaborar o Quadro 7, que apresenta a abordagem com a sua forma de utilização no ensino de programação nas instituições de nível superior na área de computação.

Quadro 7 – Forma de utilização das abordagens no ensino superior.

Abordagem	Utilização no ensino
Serious Games	Conduzir as aulas por meio de alguma ferramenta classificada como Jogo Sérió (por exemplo o Scratch) e trabalhar o conteúdo da disciplina por meio da ferramenta adotada.
Propostas ou uso de Ferramentas	Utilizam-se as ferramentas propostas durante as aulas, onde o conteúdo da disciplina é abordado com o auxílio das ferramentas, normalmente as aulas requerem o uso de laboratório.
Coding Dojo	Realizam-se sessões de Coding Dojo durante uma quantidade pré-determinada de aulas, problemas relacionados ao conteúdo da disciplina são apresentados para que os alunos resolvam.
Sala de Aula Invertida	Primeiramente, adotam-se plataformas que permitam a interação remota entre professores e alunos, os conteúdos da disciplina são trabalhados extraclasse, e na sala de aula o professor administra a socialização do conhecimento. Também pode-se realizar encontros presenciais que incluam atividades como: atividades interativas, incluindo pequenas palestras e desafios de programação.
Gamificação	A metodologia atribui recompensas aos alunos e com isso, permitindo formar um ranking com essas pontuações.
Ensino Híbrido	O ensino híbrido é composto por aulas presenciais e por aulas a distância. Nesse sentido, utilizou-se uma ferramenta (CodeBench) que permitiu auxiliar as aulas remotas.

Abordagem	Utilização no ensino
Desafios de Programação	Realização periódica de desafios/competições de programação, pode-se utilizar problemas de maratonas oficiais de programação, que abordem os conteúdos da disciplina e de juízes eletrônicos para corrigir esses problemas.
Método Focado em Exercícios	Aulas realizadas em laboratório, com dois momentos, a) ensino teórico dos conteúdos da disciplina e b) aplicação de exercícios práticos. Periodicamente realizam-se avaliações parciais.
Dispositivos Móveis	Aulas em laboratórios mediadas pelo uso de dispositivos móveis, o conteúdo da disciplina é abordado em um primeiro momento em seguida as tarefas são realizadas com o uso de dispositivos móveis.
Redes Sociais	Aplicação de exercícios extraclasse (por alguma rede social pré-determinada) periodicamente com o objetivo de desenvolver o hábito de programação, o feedback dos alunos, bem como suas dúvidas também pode ser obtido pelas redes sociais.
Ambiente Virtual	Seleciona-se um ambiente virtual, explica-se os objetivos da utilização, bem como o correto manuseio, após este processo, as aulas começam a ser realizadas com o apoio da ferramenta, atividades presenciais e extraclasse podem ser realizadas por meio do ambiente virtual
Programação em Tempo Real	Ensinar a programação escrevendo código real/ao vivo durante as aulas. Periodicamente, realizar sessões de codificação ao vivo, no formato de palestras.
<i>Problem Based Learning - PBL</i>	Instigar por meio de questões-problema, o aprendizado constante e autônomo dos alunos, fazendo com que os mesmos entrem em uma rotina de estudos constantes, ao serem provocados a entregar tarefas periodicamente.
<i>Massive Open Online Course - MOOC</i>	Gravação de vídeos com explicações sobre os conteúdos abordados; a aplicação de exercícios e de testes para avaliar o aprendizado dos alunos; Disponibilização dos materiais de aula em uma plataforma online.
Arduino	Os alunos utilizam kits de Arduino para trabalhar os conceitos da disciplina. O ensino foi dividido em dois módulos: 1) Tutoriais com crescente dificuldade de prática e 2) Aprofundamento na linguagem de programação através de expressões aritméticas e estruturas condicionais e de controle.

Fonte: Elaboração própria (2020).

5.2.5 QS4: Como foi feita a avaliação do processo de ensino e aprendizagem de programação com a utilização as abordagens identificadas na pesquisa?

Com base na QS4, realizou-se a extração da forma de avaliação utilizada nas abordagens identificadas nos trabalhos, conforme ilustra o Quadro 4. É importante ressaltar que foi realizada uma síntese nas formas de avaliações apresentadas nos artigos. Com essa extração de informações podemos elencar uma série de boas práticas para a avaliação de uma determinada abordagem.

Quadro 8 – Forma de avaliação utilizada nas abordagens

Abordagem	Forma de avaliação
Serious Games	Observação das aulas laboratoriais visando analisar o desempenho dos alunos, questionários avaliativos online, bem como entrevistas semiestruturadas.
Propostas ou uso de	Provas de nivelamento do conhecimento, listas de exercícios, provas

Abordagem	Forma de avaliação
Ferramentas	finais e questionários de avaliação.
Coding Dojo	Aplicação de questionários e observação do desempenho durante as sessões
Sala de Aula Invertida	Aplicações de questionários de autoavaliação e provas que abordem os conteúdos da disciplina.
Gamificação	Avaliação por meio de dinâmicas, mecânicas e componentes. Como dinâmicas pode-se listar: Progresso, Regras e Relacionamentos; As mecânicas podem ser: Competições e Cooperações, Desafios, Feedback e Recompensas; Como componentes, lista-se: Emblemas, Missões, Pontos, Níveis ou Fases e Tabelas de Classificação ou Ranking.
Ensino Híbrido	Avaliação por meio de análises qualitativas, a partir de dados obtidos por meio de questionários, os quais são aplicados com o intuito de verificar o entendimento e as dificuldades enfrentadas pelos alunos, com relação a metodologia utilizada.
Desafios de Programação	Avaliação baseada na análise do desempenho dos alunos antes da utilização da metodologia, em comparação com o desempenho obtido, ao utilizar a metodologia. Os problemas dos desafios podem ser validados por meio de juízes eletrônicos.
Método Focado em Exercícios	As questões da avaliação normalmente eram adaptações de questões trabalhadas no laboratório de codificação. O desempenho dos alunos na avaliação parcial contribuía em maior peso para a média final na disciplina.
Dispositivos Móveis	Tarefas realizadas com o dispositivo móvel, avaliação de satisfação com questionários que adotam a escala de likert.
Redes Sociais	Autoavaliação através de questionários, buscando identificar: melhorias significativas relacionadas a satisfação, auto confiança e engajamento no estudo de programação.
Ambiente Virtual	Aplicação de tarefas utilizando o ambiente virtual
Programação em Tempo Real	Pré testes, pós testes e Coleta de feedback dos alunos.
<i>Problem Based Learning</i> - PBL	Avaliação por meio da resolução de problemas que abordem os conteúdos trabalhados na disciplina.
<i>Massive Open Online Course</i> - MOOC	Exercícios são realizados através de questionários sobre os conteúdos abordados. A efetividade da utilização também pode ser avaliada por meio de questionários online.
Arduino	Avaliação por meio de questionários de satisfação

Fonte: Elaboração própria (2020).

6 CONCLUSÕES

Com o intuito de identificar pesquisas que atendessem aos objetivos desta pesquisa, que estão relacionados a identificar quais abordagens estão sendo utilizadas atualmente no ensino de programação pelas instituições de ensino de nível superior na área de computação, uma RqSL foi executada e permitiu responder as questões de pesquisas presentes neste estudo. Para isso, selecionou-se e avaliou-se 50 artigos relevantes aos objetivos desta pesquisa.

Com a execução da RqSL conseguiu-se identificar diversas metodologias que vêm sendo adotadas atualmente no ensino de programação nos cursos de nível superior. Observou-se que as abordagens mais utilizadas na literatura analisada foram: a) *Serious Games*, b) propostas ou uso de ferramentas, c) sala de aula invertida e d) gamificação. As

abordagens apresentaram resultados positivos em suas implementações. Isto permitiu elencar uma série de informações de melhorias no processo de ensino e aprendizado de programação.

Pode-se notar que, dos 15 países identificados nesta pesquisa, o Brasil destaca-se como o país que mais produziu trabalhos relacionados à utilização de abordagens que visem a melhoria do processo de ensino e aprendizagem de programação no ensino superior, com 22 publicações no período de 2016 a 2019. Dentre as Universidades, 43 foram identificadas nesta RsQL, onde a Universidade Federal do Pará (UFPA) foi a que mais produziu trabalhos, com um total de 3 produções.

Os resultados apresentados nesta pesquisa servem de base para futuros trabalhos que pretendam desenvolver ou implementar abordagens que estimulem a melhoria do ensino e aprendizado de programação, uma vez que foram identificados e extraídos da literatura relatos de dificuldades, forma de utilização e meios de avaliar cada abordagem selecionada pela RsQL.

Diante do exposto, pretende-se continuar evoluindo esta pesquisa a fim de elaborar uma estratégia de ensino que utilize as principais abordagens que foram identificadas com a RsQL e aplicá-las em turmas de programação de universidades brasileiras, assim como realizar a avaliação dos pontos fortes e fracos da estratégia que será implementada.

REFERÊNCIAS BIBLIOGRÁFICAS

AURELIANO, V. C. O.; TEDESCO, P. C. de A. R. (2012) Ensino-aprendizagem de Programação para Iniciantes: uma Revisão Sistemática da Literatura focada no SBIE e WIE. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*.

BERSSANETTE, J. H.; FRANCISCO, A. C.; BARAN, L. R. (2018) Espaços Ampliados Apoiados Por Tecnologias Digitais Para o Ensino de Programação de Computadores. *Revista Pleiade*, v. 12, n. 25, p. 39-51.

BRASIL - Ministério da Educação. (2016) Diretrizes Curriculares Nacionais para os cursos de Computação. Brasília: MEC. Disponível em: < <http://portal.mec.gov.br/busca-geral/323-secretarias-112877938/orgaos-vinculados-82187207/12991-diretrizes-curriculares-cursos-de-graduacao> >

PARSONS, D; WOOD, K.; HADEN, P. (2015) What are we doing when we assess programming?. In *17th Australasian Computing Education Conference (ACE 2015)*. Australian Computer Society, Inc, 119–127. <http://crpit.com/confpapers/CRPITV160Parsons.pdf>

GIRAFFA, L. M. M.; MORAES, M. C. (2013) Evasão na Disciplina de Algoritmo e Programação: Um Estudo a Partir dos Fatores Intervenientes na Perspectiva do Aluno. In: *Tercera Conferencia sobre el Abandono en la Educación Superior (III CLABES)*, México.

GONÇALVES, B. et al. (2019) Elementos de Gamificação Aplicados no Ensino-Aprendizagem de Programação Web. In: *Anais do XXVII Workshop sobre Educação em Computação*. SBC, 2019. p. 1-10.

GROTTA, A. (2019) Aprendizagem baseada em projeto ágil para educação em programação de computadores no ensino superior brasileiro. 2019. 154 p. Dissertação (Mestrado em Ciências) - *Escola de Artes, Ciências e Humanidades*, Universidade de São Paulo, São Paulo.

HIJON-NEIRA, R. B.; VELÁZQUEZ-ITURBIDE, Á.; PIZARRO-ROMERO, C. (2014). Game programming for improving learning experience. *In: Proceedings of the 2014 conference on Innovation & technology in computer science education.*, pp. 225-230.

KITCHENHAM, B.; Charters, S. (2007). Guidelines for performing Systematic Literature Reviews. *In Software Engineering. (3^o ed., Vol. //).* -2007-01,: EBSE Technical Report.

LIU, Y.; TONG, Y.; YANG, Y. (2018) The Application of Mind Mapping into College Computer Programming Teaching. *Procedia Computer Science*, v. 129, p. 66-70.

LUXTON-REILLY, A. et al. (2019) Pass Rates in Introductory Programming and in other STEM Disciplines. *In: Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education.* p. 53-71.

MALCHER, P. R. C.; FERREIRA, D. A. L.; OLIVEIRA, S. R. B.; VASCONCELOS, A. M. L. (2015) Um Mapeamento Sistemático sobre Abordagens de Apoio à Rastreabilidade de Requisitos no Contexto de Projetos de Software. *Revista de Sistemas de Informação da FSMA* n. 16, pp. 3-15.

MARINHO, C. et al. (2016) Experiências no uso da metodologia coding dojo nas disciplinas básicas de programação de computadores em um curso interdisciplinar do ensino superior. *In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação.* p. 1097.

MEDEIROS, T. J.; DA SILVA, T. R.; ARANHA, E. H. S. (2013) Ensino de programação utilizando jogos digitais: uma revisão sistemática da literatura. *RENOTE-Revista Novas Tecnologias na Educação*, 11(3).

NASCIMENTO, A. C. S. et al. (2018) Propriedades Psicométricas do Action Research Arm Test: Revisão Sistemática da Literatura.

NOGUEIRA, F. et al. (2020) PBL e robótica no ensino de conceitos de Lógica de Programação. *In: Anais do XXIV Workshop sobre Educação em Computação.* SBC. p. 298-307.

QUARESMA, J. A. S.; ELIASQUEVICI, M. K.; BEZERRA, S. R. O.; MENEZES, J. (2019) Abordagens gamificadas para o ensino e aprendizagem na disciplina Algoritmos: uma Revisão Sistemática da Literatura. *In: 16th CONTECSI - International Conference on Information Systems and Technology Management*, São Paulo.

RAMOS, V. et al. (2015) A Comparação da Realidade Mundial do Ensino de Programação para Iniciantes com a Realidade Nacional: Revisão sistemática da literatura em eventos brasileiros. *Simpósio Brasileiro de Informática na Educação*, p. 318 – 327.

SILVA, T. R. et al. (2015) Ensino-aprendizagem de programação: uma revisão sistemática da literatura. *Revista Brasileira de Informática na Educação*, v. 23, n. 01, p. 182.

SPRINT, G.; COOK, D. (2015). Enhancing the CS1 student experience with gamification. *In Integrated STEM Education Conference (ISEC)*, IEEE (pp. 94-99). IEEE.

VIEIRA, C. E. C.; LIMA JÚNIOR, J. A. T.; VIEIRA, P. P. (2015) Dificuldades no Processo de Aprendizagem de Algoritmos: uma Análise dos Resultados na Disciplina de AL1 do Curso de Sistemas de Informação da FAETERJ–Campus Paracambi. *Cadernos UniFOA*, v. 10, n. 27, p. 5-15.

WATSON, C.; LI, F. W. B. (2014) Failure rates in introductory programming revisited. *In Proceedings of the 2014 conference on Innovation & technology in computer science education (ITiCSE '14)*. ACM, New York, NY, USA, 39-44.738.