Metodologias de Ensino na Programação Paralela com Placas Gráficas: Uma Revisão Sistemática da Literatura

Roberto Almeida Jr, Hadria Farias, Marcos Saavedra, Josivaldo Araújo

[1]Instituto de Ciências Exatas e Naturais – Universidade Federal do Pará (UFPA)
Rua Augusto Corrêa, 01 – 66.075-110 – Belém – PA – Brasil

qbetoo, hadriafarias, marcosaavedraa3{@gmail.com}, josivaldo@ufpa.br

Abstract. *Parallel programming teaching has become a current challenge in educational institutions for requiring not only its own infrastructure — not always differentiated methods and software of sequencial programming. Such task is even more challenging when graphics cards are involved. This present paper carries out a Systematic Review of the Literature and it has objective at identifying methodologies that can help in the teaching-learning process of parallel programming using graphics cards (GPGPU). Searches were carried out in national and international bases resulting in 445 returned studies, from which only 2 were select by inclusion criteria.*

Resumo. *O ensino da programação paralela vêm sendo um desafio atual e crescente nas instituições de ensino, pois requer, além de uma infraestrutura própria que nem sempre está disponível, métodos e softwares diferenciados da programação sequencial. E essa tarefa se mostra ainda mais desafiadora quando envolve o uso de placas gráficas. Este trabalho apresenta uma Revisão Sistemática da Literatura com o objetivo de identificar metodologias que possam auxiliar no processo de ensino-aprendizagem da programação paralela com o uso de placas gráficas (GPGPU). Foram realizadas pesquisas em bases nacionais e internacionais que resultaram em 445 estudos retornados, sendo que destes, apenas 2 estudos foram selecionados nos critérios de inclusão.*

1. Introdução

A Computação de Alto Desempenho (CAD) proporcionou um grande avanço na forma de se processar informações e dados nas últimas décadas. O projeto dos supercomputadores, que passaram a contar não apenas com esses poderosos processadores, mas também, com unidades de processamento gráfico, fez com que a pesquisa evoluísse bastante, pois permitiu apresentar soluções cada vez mais rápidas, em curtos intervalos de tempo, independente da complexidade dos problemas [Chen et al. 2018]. A forma de se programar foi modificada nesses ambientes, pois se passou de uma programação estritamente sequencial, para uma programação onde se procura obter ganhos de desempenho, com níveis elevados de paralelismo, não apenas em hardware, mas também, na elaboração dos códigos-fontes [Bachiega et al. 2018].

Apesar da programação paralela já está consolidada nos grandes centros de pesquisa e nos projetos dos supercomputadores, parece que ainda está em um processo de evolução nos cursos de graduação em Computação, principalmente, quando o ensino da programação paralela, tem como objeto de estudo, as placas gráficas ou GPGPU (*General*

DOI: 10.5753/sbie.2021.218451

Purpose Graphics Processing Unit), e isso não apenas no Brasil, mas também, em alguns países do mundo, fazendo com que haja um grande descompasso entre a pesquisa e o ensino [Chen et al. 2018].

O ensino da Programação Paralela é uma das competências que deve ser desenvolvida nos eixos de formação nos cursos de graduação em Computação, e por conta disso, faz parte das recomendações nas diretrizes curriculares para a área, propostas por importantes instituições, como a ACM (*Associatioin for Computing Machinery*) e o IEEE (*Institute of Electrical and Eletronics Engineers*) [ACM and IEEE-CS 2020], no contexto mundial, e pela SBC (Sociedade Brasileira da Computação) [Zorzo et al. 2017], no cenário nacional.

Essa lenta evolução no ensino da programação paralela pode ser resultado dos constantes desafios impostos, que começam com a plataforma computacional necessária para o ensino, passando pelos modelos de programação e diferentes formas de se comunicar e sincronizar, chegando a formação dos alunos de Computação, na graduação. E para superar isso, é necessário instituir um ”pensamento paralelo”, fazendo com que os alunos desenvolvam competências e habilidades que os proporcionem trabalhar com facilidade nesses ambientes computacionais [Bachiega et al. 2018]. Dessa forma, este trabalho apresenta uma Revisão Sistemática da Literatura (RSL) que têm como objetivo identificar metodologias que possam auxiliar no processo de ensino-aprendizagem da programação paralela com o uso de placas gráficas (GPGPU).

2. Trabalhos Relacionados

Há na literatura poucos trabalhos secundários que retratam o ensino da programação paralela, no entanto, não foram identificados estudos que tratassem, especificamente, com o uso de placas gráficas. Entre os estudos secundários identificados, pode-se citar o trabalho de [Bachiega et al. 2017], onde foi realizado um Mapeamento Sistemático da Literatura (MSL) considerando o ensino teórico e prático da programação paralela. Foram analisados métodos de aprendizagem, configuração de software e infraestrutura de laboratórios. O estudo apresentou, como principal contribuição, uma visão do estado da arte, destacando as principais práticas de ensino envolvidas, no período de 2006 a 2017. No entanto, dos 24 estudos apresentados, apenas três trabalhos indicavam o uso de placas gráficas.

Já em seu segundo trabalho [Bachiega et al. 2018] apresenta um estudo do ensino da programação paralela em 45 universidades brasileiras, entre públicas e privadas. Entre essas, 4 não apresentaram disciplinas relacionadas com paralelismo em sua matriz curricular. O conteúdo de programação paralela aparece em 10 diferentes nomes de disciplinas, no entanto, o que chama atenção é à carga horária prática, que em muitos casos é inferior, e em outros, inexistentes, quando comparadas a carga horária teórica. Ou seja, percebe-se que os conteúdos ministrados são minimamente praticados.

No trabalho de [Saeed 2020], é apresentada uma pesquisa voltada a professores de 100 universidades que lecionam, ou tiveram algum contato com o ensino da linguagem CUDA. O objetivo foi avaliar se a linguagem é adequada para o ensino dos princípios de computação paralela, bem como, a relevância para se criar cursos específico nas universidades dos EUA. Através de formulários e entrevistas, o autor identificou que mais 70% dos participantes apontaram que a linguagem CUDA deve ser usada para o ensino da programação paralela.

3. Revisão Sistemática da Literatura

A principal meta de uma Revisão Sistemática da Literatura (RSL) consiste em realizar uma pesquisa exaustiva na literatura, em busca de evidências que possam apoiar uma determinada hipótese, ou simplesmente a busca por conhecimento aprofundado acerca de certo fenômeno de interesse [Pereira et al. 2019]. O objetivo da RSL desenvolvida neste trabalho, foi o de identificar e descrever estudos que abordem o ensino da programação paralela utilizando como prática, as placas gráficas, isso faz com que ela seja classificada como uma revisão narrativa [Galvão and Ricarte 2019]. Nas seções seguintes, são discutidas as etapas de definição do Protocolo de Pesquisa, bem como a sua execução.

3.1. Questões de Pesquisa

A finalidade de um estudo secundário é identificar estudos primários de onde se possa extrair e analisar os dados [Silva et al. 2020]. Para isso, é essencial a elaboração de uma Questão de Pesquisa (QP) que seja objetiva e precisa o suficiente, para ajudar no processo de pesquisa e extração de dados. A questão de pesquisa formulada foi:

- **Quais são as metodologias utilizadas para o ensino-aprendizagem da programação paralela utilizando placas gráficas?**

Para obter os resultados esperados, por meio da RSL, além da QP, elaborou-se Questões Secundárias (QS), que possuem o objetivo de detalhar informações que a RSL procura identificar, contribuindo, dessa forma, com a análise do estudo. As QS são apresentadas no Quadro 1:

Quadro 1. Questões Secundárias

QS	Perguntas
QS1	Quais foram os instrumentos de avaliação utilizados para verificar o aprendizado dos alunos?
QS2	Quais foram os problemas utilizados para o ensino da programação paralela com placas gráficas?
QS3	Como as atividades foram desenvolvidas no processo de ensino-aprendizagem com placas gráficas?
QS4	Quais foram as técnicas e/ou ferramentas (*cluster* físicos, placas gráficas, virtualizações, bibliotecas, entre outras) desenvolvidas ou utilizadas no processo de ensino-aprendizagem?
QS5	Qual é a caracterização da organização de ensino superior (pública, privada) onde a disciplina foi ministrada?
QS6	Quais foram as principais dificuldades relatadas, por professores e alunos, no processo de ensino-aprendizagem?

3.2. *String* de Busca

Seguindo o protocolo de pesquisa, o qual descreve todos os passos e métodos necessários para a construção da RSL, nesta etapa, definiu-se a *string* de busca, apresentada no Quadro 2, nos idiomas português e inglês. Esta foi estruturada segundo [Pereira et al. 2019], em *Population, Intervention, Context, Outcomes, Comparison* (PICOC). Entretanto, apenas os itens População, Intervenção e Resultados foram considerados relevantes para a

pesquisa. Tal restrição, caracteriza esta pesquisa como uma Revisão QUASI Sistemática da Literatura.

No entanto, nem sempre as fontes consultadas retornavam resultados. Para essas fontes foram utilizadas buscas manuais, em uma tentativa de se encontrar estudos. Foram então utilizadas as frases, em cada um dos idiomas, baseadas nas palavras constituintes das *strings* de busca, definida no Quadro 2.

Quadro 2. *Strings* Usadas

Versão	String
Português	(”programação paralela”) AND (”método”OR ”prática”OR ”técnica”OR ”metodologia”OR ”ferramenta”OR ”abordagem”) AND ((”placa gráfica”OR ”GPU”OR ”GPGPU”) AND (”ensino”OR ”aprendizagem”))
Inglês	(*”parallel programming”) AND (”method”OR ”practice”OR ”technique”OR ”methodology”OR ”tool”OR ”approach”) AND ((”graphics card”OR ”GPU”OR ”GPGPU”) AND (”teaching”OR ”learning”*))

3.3. Processo de Busca

Como o objetivo desta RSL é identificar metodologias existentes que possam auxiliar no processo de ensino-aprendizagem da programação paralela com o uso de placas gráficas, serão considerados apenas os trabalhos disponíveis nas bases na área da Educação e, em decorrência do tema, também, em bases específicas da área de Computação de Alto Desempenho, seja nacional ou internacional, como mostra a lista de fontes disponível em https://bit.ly/2UdojAt.

Para a realização deste trabalho foram considerados apenas estudos completos, publicados em periódicos ou conferências, escritos em português ou inglês. Os estudos primários foram divididos em experimentais (ou *Empirical Studies*); teóricos (estudos conceituais baseados em um entendimento de uma área); e, relatos de experiência em docência acadêmica.

3.4. Critérios de Exclusão e Inclusão

Após a utilização da *string* de busca, de forma automática e manual, nas bases de pesquisa, foram aplicados os critérios de inclusão e exclusão, definidos e apresentados no Quadro 3, para realizar a filtragem nos estudos retornados. Esses critérios servem para avaliar a qualidade do estudo, criando uma lista de trabalhos que atendem os objetivos da pesquisa, e outra que estão fora do seu escopo [Pereira et al. 2019].

3.5. Execução do Estudo Secundário

Após definido o protocolo da revisão sistemática, com seus atributos, como *strings* de busca, fontes de pesquisa, entre outras informações, cada pesquisador iniciou a busca automática em sua respectiva base. Os estudos retornados pelas bases foram armazenados no software especializado JabRef, que fornece um banco de dados para o auxílio desta tarefa. A pesquisa foi realizada considerando os últimos 5 anos, pois foi considerada as Referências de Formação para os cursos de Bacharelado em Ciência da Computação (RF-CC-17), definida pela Sociedade Brasileira de Computação (SBC) [Zorzo et al. 2017],

Quadro 3. Critérios de Exclusão e Inclusão

Tipo	Descrição do Critério
CE.1	Estudos que não estejam disponíveis livremente para consulta ou download (em versão completa) através das fontes de pesquisa ou através das ferramentas de busca Google (http://www.google.com.br/) e/ou Google Scholar (http://scholar.google.com.br/)
CE.2	Estudos que claramente não atendam as questões de pesquisa;
CE.3	Estudos repetidos (em mais de uma fonte de busca) terão apenas sua primeira ocorrência considerada
CE.4	Estudos enquadrados como resumos, *keynote speeches*, cursos, tutoriais, *workshops* e afins
CE.5	Estudos que não mencionem as palavras-chave da pesquisa no título, resumo ou nas palavras-chave do artigo, salvo trabalhos que abordem algum processo de ensino de programação paralela voltado à placas gráficas
CE.6	Estudo inserido no contexto, mas que não aborde o ensino da programação paralela utilizando placas gráficas
CE.7	Estudos não apresentados em uma das linguagens aceita (portuguesa e inglesa).
CI.1	Trabalhos que apresentem primariamente abordagens no contexto de ensino da programação paralela e que utilize as placas gráficas como parte ou totalidade da abordagem
CI.2	Trabalhos que apresentem relatos de práticas de ensino da programação paralela utilizando placas gráficas.

retornando, dessa forma, 445 estudos, como pode ser observado na Tabela 1, dos quais, apenas 2 artigos, foram selecionados após passarem pelos critérios de inclusão e exclusão. Os estudos selecionados podem ser visualizados no Quadro 4.

Quadro 4. Estudos Resultantes dos Critérios de Exclusão e Inclusão

Título	Referência
EP.1. *Activity Based Approach for Teaching Parallel Computing: An Indian Experience*.	[Chitra and Ghafoor 2019]
EP.2. *GPGPU Programming for CS Undergraduates: Which one is Superman?*	[Fenwick and Norris 2020]

3.6. Avaliação da Qualidade

A avaliação da qualidade de um artigo científico é a mensuração de sua relevância e conteúdo. Essa avaliação é um dos critérios de inclusão ou exclusão aplicados aos estudos durante a seleção. Os critérios de avaliação da qualidade dos estudos primários utilizados nesta RSL são apresentados no link a seguir: https://quadro-ava-qualidade.

Como é possível notar, os critérios (1) a (4) são genéricos, ou seja, aplicam-se a todos os estudos primários avaliados, enquanto os critérios (5) a (7) são específicos, e aplicam-se especificamente aos respectivos tipos de trabalhos mencionados. Os estudos primários selecionados foram lidos em sua totalidade, e então, foram avaliados quanto

Tabela 1. Seleção dos Estudos

Fontes	Estudos retornados	remoção repetidos	excluidos 1ª Seleção	2ª Seleção								
				Excluídos							Incluídos	
				CE1	CE2	CE3	CE4	CE5	CE6	CE7	CI1	CI2
Transactions on Computing Education	357	22	0	23	123	2	56	45	85	0	0	1
Transactions on Education	13	0	2	1	7	0	2	0	0	0	0	1
International Journal of Educational Technology in Higher Education	11	0	2	0	7	0	2	0	0	0	0	0
Principles and Practice of Parallel Programming	64	19	0	3	29	13	0	0	0	0	0	0
				27	166	15	60	45	85	0	0	2
Total	**445**	41	4	398							**2**	

aos critérios de qualidade. Para realizar a avaliação de qualidade dos estudos incluídos, foi utilizada a escala *Likert*-3, que representa a adesão destes atributos aos critérios de qualidade. Isso permite respostas gradativas: 0 (discordo totalmente), 1 (Neutro) e 2 (concordo totalmente), pois com menos itens, os critérios se tornam menos subjetivos [Pereira et al. 2019]. A escala utilizada para cada critério de qualidade pode ser acessada no link: https://quadro-escala-likert3

4. Resultados e Discussões

Nesta seção será apresentada a condução das análises dos resultados dos estudos primários, visando responder as questões de pesquisa.

4.1. Avaliação da Qualidade dos Estudos Primários

Para cada artigo avaliado foi atribuída uma pontuação, assim enquadrando-o em um dos cinco níveis de qualidade definidos por [Beecham et al. 2008]. Os níveis podem ser acessados em: https://quadro-niveis-qualidade

Realizada a análise, e após os critérios serem aplicados, restaram apenas 2 estudos. Sendo [Chitra and Ghafoor 2019] classificado como Excelente, com 91% de qualidade, e [Fenwick and Norris 2020] classificado como Boa, com 55% de qualidade.

4.2. Respostas às Questões de Pesquisa

Esta seção responde as questões de pesquisa levantadas no protocolo desta RSL como meio de investigar o estado da arte no âmbito de metodologias que abordem o ensino da programação paralela com placas gráficas. As respostas são definidas segundo os códigos apresentados no Quadro 4.

QP: Quais são as metodologias utilizadas para o ensino-aprendizagem da programação paralela utilizando placas gráficas?

Durante a etapa de extração de dados, foram selecionados dois estudos que relatam as suas experiências no processo de ensino-aprendizagem da programação paralela com

placas gráficas. Os autores utilizam diferentes metodologias, sendo que, em um dos casos, utiliza-se mais de uma prática de ensino.

No estudo [EP.1], foram utilizadas diferentes metodologias de ensino, desde aulas tradicionais expositivas; passando por discussões intermediadas pelo professor, que acolhia os temas sugeridos pelos alunos, como problemas de interesse específicos, que contribuíam para o aprofundamento dos estudos em temas como coerência de cache, paradigmas de programação paralela; chegando ao desenvolvimento de aplicativos móveis. Essas discussões resultavam em atividades de miniprojetos, os quais eram expostos como problemas aos alunos, e estes tinham que solucioná-los, em grupos, e durante o seu desenvolvimento, as ideias de implementação eram avaliadas. Porém, para se chegar ao ensino da programação paralela utilizando placas gráficas, com CUDA, passava-se por tecnologias e bibliotecas anteriores, como o uso de vários processadores, com bibliotecas como o OpenMP e MPI. No entanto, além das práticas tradicionais, foram utilizadas, atividades desplugadas e o uso da programação curta em sala de aula. As atividades desplugadas, desenvolviam-se com o objetivo de ilustrar conceitos de programação paralela sem o uso do computador. Já na prática que utiliza o uso da programação curta em sala de aula, eram repassados problemas considerados de pequena complexidade, os quais não geram muitas linhas de códigos, e com isso, era possível discuti-los, bem como, solucioná-los durante as aulas.

No estudo [EP.2] foi utilizada uma metodologia analítica, ou seja, o ensino foi dividido em duas partes: a primeira, de forma teórica, onde o conteúdo era repassado de forma tradicional, com explanação dos conceitos, seguidos de exemplos; e a segunda, de forma prática, onde os conceitos eram desenvolvidos em uma abordagem baseada em problemas e desafios, sendo esses, específicos na área de Computação Gráfica. Ao longo do curso, os alunos iam solucionando pequenas partes de um único problema, que no caso, era realizar a identificação de uma imagem. E as etapas desse problema, eram ordenados em um crescente grau de complexidade.

QS.1: Quais foram os instrumentos de avaliação utilizados para verificar o aprendizado dos alunos?

Em [EP.1] o professor, para avaliar o desempenho e o nível de aprendizado dos alunos, assim como, o desempenho geral do curso, para fins de comparação com os anos anteriores, utilizou uma pesquisa usando escala *Likert*-4, avaliando, dessa forma, a compreensão dos alunos por meio de testes curtos em sala de aula. Foram também utilizados, questionários para verificar o entendimento dos tópicos ministrados, levando em consideração o tema a ser implementado. Além disso, eram realizadas avaliações através de códigos de fácil entendimento, e que eram solicitadas a saída para estes. Eles tinham alguns minutos para pensar e responder. As respostas eram discutidas em sala, sendo fornecida a solução, com explicação, pelo professor.

No estudo [EP.2] é citado, como forma de avaliação, uma única atividade avaliativa prática que era realizada no final do curso, mas que levava em consideração as etapas desenvolvidas durante o curso. O resultado da avaliação não é citado no trabalho, no entanto, a mesma é disponibilizada pelos autores.

QS.2: Quais foram os problemas utilizados para o ensino da programação paralela com placas gráficas?

No trabalho [EP.1], as aplicações envolviam varredura de parâmetros, simulação da corrente nos oceanos, filtragem de ruído em imagens, aplicações do algoritmo Gauss-Seidel para sistemas lineares e aplicações de classificação do tipo ímpar-par, sendo estes últimos, por suas estruturas, melhores solucionados por meio da programação paralela, beneficiando-se dos vários núcleos das placas gráficas.

Já o estudo [EP.2], desenvolveu o curso em um problema de identificação de imagem, através da comparação, cuja tarefa é categorizá-las em uma série de classes pré-definidas de acordo com seus conteúdos. Os tópicos do curso foram ministrados através da resolução deste problema, sendo este, implementado de forma gradual. O problema consiste no manuseio dos *pixels* (pontos de coloração de telas) que são processados de forma paralela, com o uso de placas gráficas.

QS.3: Como as atividades foram desenvolvidas no processo de ensino-aprendizagem com placas gráficas?

Em [EP.1] o ensino foi desenvolvido em uma disciplina específica, denominada de Computação Paralela e Distribuída, como parte do curso de pós-graduação em Ciência da Computação. São descritas diversas atividades, que foram realizadas tanto de forma individual, como em equipes. Exercícios teóricos eram aplicados em sala, de forma individual, assim como, trabalhos de programação curta. Atividades práticas, eram realizadas em duplas. E ao final, o miniprojeto foi realizado em equipes de até três integrantes.

No [EP.2], o ensino também foi desenvolvido em uma disciplina específica, chamada de Uma Introdução para GPGPU usando CUDA e foi realizada em um curso de verão, como uma extensão da disciplina Introdução aos Sistemas de Computadores, que faz parte da grade fixa da graduação em Ciência da Computação. Esta disciplina foi ministrada para alunos de variados níveis de escolaridade (técnico, graduação e pós-graduação). Tanto a prática, quanto a teoria, foram trabalhadas de forma individual. O estudo descreve a primeira etapa do curso como teórica, porém o foco descrito no artigo é na parte prática, a qual eram desenvolvidas etapas de implementação, com base em cada tópico teórico inicialmente visto.

QS.4: Quais foram as técnicas e/ou ferramentas (cluster físico, placa gráfica, virtualização, bibliotecas) desenvolvidas ou utilizadas no processo de ensino-aprendizagem?

Em [EP.1] os alunos tinham uma introdução aos conceitos básicos de aplicações utilizando OpenMP e MPI, para assim, iniciarem os conceitos em CUDA. É descrito que os alunos utilizavam laptops e PCs com GPUs Nvidia para as aulas práticas, porém não fornecendo suas respectivas configurações técnicas.

O estudo [EP.2] descreve o uso exclusivo de placas gráficas GPGPU Nvidia. Para isso, suas implementações eram realizadas utilizando a API OpenCL e CUDA, todos os códigos foram desenvolvidos na linguagem C/C++. O hardware utilizado para o curso era um Intel Core i7-7700K com uma única placa de vídeo NVIDIA GeForce GTX 1080. O sistema operacional era Ubuntu Linux, no entanto, não foi relatada a versão.

QS.5: Qual é a caracterização da organização de ensino superior (pública, privada) onde a disciplina foi ministrada?

No trabalho [EP.1], o estudo é realizado na Faculdade pública Autônoma de *Thi-*

agarajar College of Engineering, na ÍNDIA, a qual é financiada, tanto pelo governo federal, quanto pelo governo estadual de Tamilnadu, cidade onde fica a Faculdade. E em [EP.2], o relato de experiência é desenvolvido na Universidade particular *Appalachian State University (App State*, nos EUA), e tem como um dos objetivos, fixar o curso na grade curricular da graduação de Ciência da Computação.

QS.6: Quais foram as principais dificuldades relatadas, por professores e alunos, no processo de ensino-aprendizagem?

Em [EP.1], os autores relatam que no processo de ensino, os alunos tiveram dificuldades em decidir os modelos mais apropriados das técnicas de programação paralela durante suas avaliações. Relatam, também, que tiveram dificuldades em relação a detalhes específicos do conteúdo dado no curso, como apontar documentos recentes sobre o problema do compartilhamento do protocolo de coerência de cache, por exemplo. Foram relatados problemas, ainda, dos alunos não possuírem conhecimentos sobre conceitos de Computação Paralela e Distribuída (CPD) antes de fazerem o curso, por não o terem adquirido na graduação, ou por nesta, os conceitos terem sido repassados de forma básica. Em [EP.2] relata que, uma das principais dificuldades para o ensino é encontrar na literatura, compartilhamento de experiências em trabalhos, sobre o ensino da programação paralela, mas especificamente sobre placas gráficas. Os autores desenvolveram a disciplina, tomando como base o livro *Programming Massively Parallel Processores* de David B. Kirk e Wen-mei W. Hwu, pois o mesmo disponibiliza todo o material necessário (slides de aulas, exercícios e avaliações) para conduzir o curso. Após questionário passado aos alunos, os autores relataram variar o nível de dificuldades que os alunos tiveram, muito a depender do conhecimento prévio do assunto. Ao final do trabalho, os autores, também expõem, que a maioria dos alunos que participaram do curso ainda não encontraram oportunidades de reutilizar as habilidades adquiridas durante a disciplina.

Os relatos descritos em [EP.1], com relação ao aprendizado, não são relatados, de maneira clara, as dificuldades sentidas pelos alunos, sendo unicamente pontuado, o fato de que alguns alunos, por possuírem "carência" nos conteúdos básicos que envolvem a programação paralela, acabaram apresentando uma maior dificuldade na realização das tarefas. Já no [EP.2], destaca alguns comentários relatados pelos alunos e, entre eles, pode-se apontar: a maior complexidade do CUDA na construção de códigos paralelos, como também, do contato inicial dos conceitos de programação paralela ter sido realizado diretamente com CUDA, ao invés de bibliotecas mais simples, como OpenMP e MPI, o que teria ajudado, segundo os alunos, no processo de aprendizado.

5. Conclusão

Neste artigo foram apresentados os estudos encontrados resultante de uma Revisão Sistemática da Literatura quanto ao ensino da programação paralela com placas gráficas (ou GPGPU). Após a aplicação do protocolo, resultaram apenas dois estudos. Percebe-se dessa forma, que o tema ainda é pouco explorado, inclusive no Brasil, já que os estudos encontrados, forma publicados por professores de instituições de ensino no exterior. Nos estudos retornados, percebeu-se as dificuldades encontradas pelos alunos em compreender e absorver os conteúdos, ou seja, a dificuldade no chamado "pensamento paralelo", e isso pode se dar, por conta da ausência de iniciativas que procurem antecipar o ensino de programação paralela para os primeiros períodos nas grades curriculares, assim como, da

ausência de estudos que abordem a interdisciplinaridade da programação paralela. Isso acaba tendo reflexos nos estudos mais avançados da programação paralela, como o uso das placas gráficas. Como trabalhos futuros, pode-se propor conteúdos e metodologias de ensino que possam auxiliar os alunos, na construção do pensamento paralelo, fazendo com que possam desenvolver as competências e as habilidades necessárias ao bom uso da programação paralela, com as placas gráficas.

Referências

ACM and IEEE-CS (2020). Computing curricula 2020: Paradigms for global computing education. *ACM IEEE-CS*.

Bachiega, N., de Souza, P. L., Bruschi, S., and Souza, S. (2017). Mapeamento sistemático do ensino teórico e prático de programação paralela. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 6, page 1089.

Bachiega, N., de Souza, P. S. L., Bruschi, S. M., and Souza, S. d. R. S. (2018). Um panorama do ensino de programação paralela e distribuída em universidades brasileiras. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, pages 480–489.

Beecham, S., Baddoo, N., Hall, T., Robinson, H., and Sharp, H. (2008). Motivation in software engineering: A systematic literature review. *Information and software technology*, 50(9-10):860–878.

Chen, J., Shen, L., Yin, J., and Zhang, C. (2018). Parallel programming course development based on parallel computational thinking. In *Proceedings of ACM Turing Celebration Conference-China*, pages 103–109.

Chitra, P. and Ghafoor, S. K. (2019). Activity based approach for teaching parallel computing: An indian experience. In *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 290–295.

Fenwick, G. and Norris, C. (2020). Gpgpu programming for cs undergraduates: Which one is superman? In *Proceedings of the 2020 ACM Southeast Conference*, pages 2–9.

Galvão, M. C. B. and Ricarte, I. L. M. (2019). Revisão sistemática da literatura: conceituação, produção e publicação. *Logeion: Filosofia da informação*, 6(1):57–73.

Pereira, E. F. O. d., de Melo, W. S., de Souza Araújo, J., Oliveira-Colaborador, S. R. B., et al. (2019). Revisão sistemática da literatura na computação forense: Um estudo de caso aplicado na recuperação de dados em mídias digitais. In *16th CONTECSI-International Conference on Information Systems and Technology Management*.

Saeed, I. A. (2020). Why cs departments should consider offering cuda as a standalone course. *Journal of Computing Sciences in Colleges*, 36(4):51–58.

Silva, S. M., Matos, G. S., Araújo, F. P. O., and de Araújo, J. S. (2020). O uso de realidade aumentada aplicada às disciplinas de ciências exatas no ensino médio brasileiro: Um mapeamento sistemático da literatura. In *Anais do XXVI Workshop de Informática na Escola*, pages 429–438. SBC.

Zorzo, A., Nunes, D., Matos, E., Steinmacher, I., Leite, J., Araújo, R., Correia, R., and Martis, S. (2017). Referenciais de formação para os cursos de graduação em computacão. *Sociedade Brasileira de Computação (SBC)*.