

Um Mapeamento Sistemático da Literatura sobre o Ensino e Aprendizagem da Programação Paralela com o uso de Placas Gráficas

Roberto B. de Almeida Jr¹, Hádria de W. Santos Farias¹, Josivaldo de S. Araújo¹

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

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

***Abstract:** Despite being a reality in large research centers, parallel programming still presents a great difference in relation to teaching in Computing undergraduate program. This difference is even more visible when parallel programming is presented with graphics cards. Moreover, the challenges posed to teaching parallel thinking are even more complex. This paper presents a systematic mapping of the methodologies used in the teaching-learning process of parallel programming and graphics cards. The methodologies identified in the five selected studies range from traditional teaching and lectures classes to the use of didactics without using a computer.*

***Resumo:** A programação paralela apesar de ser uma realidade nos grandes centros de pesquisa ainda apresenta um grande descompasso com relação ao ensino nos cursos de graduação em Computação. E essa diferença se apresenta ainda mais visível quando a programação paralela é apresentada com as placas gráficas, pois os desafios impostos ao ensino do pensamento paralelo se mostram ainda mais complexos. Este trabalho apresenta um mapeamento sistemático das metodologias e ferramentas utilizadas no processo de ensino-aprendizagem da programação paralela com placas gráficas. As metodologias apontadas nos cinco estudos selecionados vão desde o ensino tradicional com aulas expositivas até o uso de didáticas sem o uso do computador.*

1. Introdução

Os supercomputadores ganharam visibilidade nos últimos anos, nas mais diversas áreas do conhecimento, por proporcionarem soluções de problemas complexos e que necessitam de grande poder de processamento em um curto intervalo de tempo. Isso comprova que a programação paralela é uma realidade consolidada nos grandes centros de pesquisa. Porém, quando se analisa a programação paralela nos cursos de graduação em Computação, percebe-se que essa realidade ainda está em evolução, notoriamente, quando o ensino da programação paralela tem como objetivo o uso de placas gráficas [Fenwick and Norris 2020].

Esse descompasso entre ensino e pesquisa pode ser resultado da forma como os cursos de graduação em Computação abordam o tema da programação paralela. Existe um consenso de que o conteúdo sobre paralelismo deve ser distribuído ao longo da matriz curricular, fazendo com que o aluno possa construir o chamado “pensamento paralelo”

de forma contínua, o que facilitaria o desenvolvimento das competências e habilidades necessárias para trabalhar em ambientes computacionais dessa natureza [Bachiega et al. 2018]. Além disso, este convívio contínuo com o paralelismo, desde os primeiros períodos do curso, faria com que os alunos pudessem visualizar esses conceitos como uma parte natural e comum do ensino da programação, da mesma forma como acontece com a chamada programação sequencial, e não como um conteúdo avançado, de difícil compreensão e distante de sua realidade [Soares et al. 2019].

Apesar da sua lenta implantação, o ensino da programação paralela é uma das competências que faz parte dos eixos de formação dos cursos de graduação em Computação e é recomendada nas matrizes curriculares para a área, constando nas propostas de importantes instituições nacionais [Bachiega et al. 2018], como SBC (Sociedade Brasileira de Computação) [Zorzo et al. 2017], e internacionais, como a ACM (*Association for Computing Machinery*) e o IEEE (*Institute of Electrical and Electronics Engineers*) [ACM and IEEE-CS 2020].

Na literatura, o ensino da programação paralela é amplamente discutido, seja nas práticas realizadas, seja nas metodologias utilizadas [Silva et al. 2021]. No entanto, há um número reduzido de estudos que abordem ou relatem experiências da programação paralela, tendo como objeto principal de ensino, as placas gráficas ou GPUs (*Graphics Processing Unit*).

O objetivo deste trabalho é apresentar um mapeamento sobre as metodologias que podem ser utilizadas no ensino da programação paralela utilizando placas gráficas, assim como, estimular a necessidade do debate sobre o tema e orientar outros estudos que possam reportar os desafios e dificuldades encontradas. Para isso, foi realizado um Mapeamento Sistemático da Literatura (MSL) onde são apontadas as principais metodologias utilizadas, os principais conteúdos, bibliotecas e linguagens, e os principais desafios encontrados nos processos de ensino-aprendizagem.

2. Trabalhos Relacionados

Estudos reportando metodologias de ensino que podem ser aplicadas à programação nos cursos de Computação não são recentes. Muitos relatam, também, as experiências com programação paralela, porém, um número reduzido de trabalhos discute o ensino da programação paralela com placas gráficas. Talvez seja pelos constantes desafios que são impostos por esse assunto, que podem iniciar com o tipo de plataforma computacional necessária, passando pela construção do pensamento paralelo e chegando nas diferentes formas de se comunicar e sincronizar etapas.

Apesar do trabalho de Saeed (2020) não ser um mapeamento ou uma revisão sistemática da literatura, apresenta uma relevância no assunto, pois é apresentada uma pesquisa realizada com 100 professores de várias universidades que lecionam, ou que já lecionaram, ou que já tiveram algum contato relacionado ao ensino de CUDA. O objetivo da pesquisa foi identificar as vantagens e as dificuldades no ensino desta extensão nas abordagens e nos principais conceitos da programação paralela. A pesquisa também foi utilizada para se identificar a relevância em se criar cursos específicos desta natureza nas universidades americanas. Como uma das conclusões da pesquisa, que foi realizada através de formulários e entrevistas, foi identificado que mais de 70% dos que participaram apoiaram o uso do CUDA como uma ferramenta no ensino avançado da programação paralela.

No trabalho de Soares *et al.* (2019) foi realizado um mapeamento sistemático da literatura com o objetivo de analisar as produções científicas referentes ao ensino-

aprendizagem, tanto de forma presencial, como à distância, da programação paralela nas graduações na área de Computação. Para isso, foram utilizadas três bases: ACM, IEEE e *Science Direct*, além da ferramenta de busca do *Google Scholar*. Foram encontrados um total de 279 artigos, onde após a aplicação dos critérios de inclusão e exclusão, restaram 83 artigos. O trabalho discute a programação paralela em termos gerais, (semestre em que a disciplina é ministrada, linguagens utilizadas para o ensino, API/IDE, Bibliotecas, Padrões, Modelos e Plataformas utilizadas para o ensino) tanto a nível de CPU, quanto de GPU. Porém, apenas listam os trabalhos encontrados, sem detalhar metodologias, ou aprofundar discussões sobre o uso das placas gráficas.

Em um outro mapeamento sistemático da literatura, realizado por Bachiega *et al.*, (2017), foram selecionados 24 trabalhos onde foram apontadas as ferramentas e os métodos utilizados para o ensino. Um ponto relevante deste mapeamento é que dos trabalhos incluídos, apenas três, citam o uso de conceitos ligados às placas gráficas. Por este trabalho, pode-se observar que a programação paralela pode até estar presente nos cursos de graduação em Computação das universidades brasileiras, no entanto, o uso das placas gráficas, como uma ferramenta de ensino e aprendizagem da programação paralela ainda não é uma realidade em muitos cursos.

Diferente dos trabalhos apresentados que desenvolveram as suas pesquisas sobre a programação paralela de uma forma geral, este trabalho desenvolveu as suas questões de pesquisa voltadas para o ensino da programação paralela, mas com exclusivamente o uso de placas gráficas em importantes bases de pesquisa como a ACM, IEEE, entre outras. Dessa forma, a abordagem procura identificar métodos e metodologias voltadas para essa finalidade, além de apontar as principais dificuldades e os resultados relatados sobre o tema proposto.

3. Diretrizes para o Ensino da Programação Paralela

O ensino da programação sequencial, apesar de sua grande relevância, não representa sozinho a realidade da matriz curricular nos cursos de graduação em Computação. Uma dificuldade que vem sendo muito discutida, é como ensinar lógica de programação de forma a motivar o aluno a aprender, ao mesmo tempo em que ele se prepara para a utilização de ferramentas de codificação mais avançadas, aplicando conceitos de lógica de programação na prática [Silva Filho and Coutinho 2021]. E essa dificuldade aumenta, principalmente, quando são introduzidos conceitos e paradigmas da Computação Paralela.

Com o objetivo de padronizar e organizar o ensino da Computação no mundo, importantes instituições como a ACM (*Association for Computing Machinery*) e a IEEE-CS (*Institute of Electrical and Electronics Engineers - Computer Society*), uniram-se e passaram a estabelecer diretrizes curriculares internacionais, inicialmente apenas para os cursos de graduação em Ciência da Computação, e atualmente, para todos os cursos de Computação [ACM and IEEE-SC 2020]. As diretrizes propostas destacam dezoito áreas do conhecimento para o curso de Computação, entre elas, está a Programação Paralela e Distribuída.

No Brasil, os cursos de Computação são regulamentados na Resolução N° 5, de 16 de novembro de 2016 do Ministério da Educação [MEC 2016], na qual são instituídas as Diretrizes Curriculares Nacionais (DCN) para os cursos de graduação na área de Computação, e o ensino da Programação Paralela e Distribuída aparece no item VI, do parágrafo 2° do Art. 5°.

Além desta Resolução, o ensino da Computação no Brasil, também possui como referência, as diretrizes elaboradas pela Sociedade Brasileira em Computação (SBC), que através das Referências de Formação para os Cursos de Graduação em Computação [ZORZO et. al. 2017] define, além dos principais eixos de formação, também, as competências e habilidades esperadas, bem como, os conteúdos específicos a serem desenvolvidos nos cursos de Computação no Brasil. Assim como a ACM e a IEEE-CS são as referências internacionais para o ensino da Computação, a SBC é a referência nacional.

Nas diretrizes definidas pela SBC, o ensino da Programação Paralela e Distribuída aparece permeado em diferentes disciplinas: Sistemas Distribuídos, Processamento Paralelo, Arquiteturas Paralelas de Computadores e Programação Paralela e Distribuída.

3.1. Placas Gráficas como Ferramentas de Aprendizagem na Programação Paralela.

As placas gráficas já nasceram com uma concepção paralela, pois eram formadas por dezenas de núcleos e tinham a finalidade de melhorar o processamento gráfico, principalmente durante a execução de jogos. No entanto, a comunidade científica observou que esses núcleos, além das funções gráficas, poderiam ser utilizados para processar dados de forma geral [Gardner 2017].

Em 2007, essa percepção se concretizou como uma revolução nos ambientes da Computação de Alto Desempenho, pelo lançamento do CUDA¹ (*Compute Unified Device Architecture*), pela NVIDIA². Dessa forma, foi possível desenvolver códigos-fontes, utilizando paralelismo de dados, para solucionar problemas das mais variadas áreas do conhecimento. CUDA C/C++ é uma extensão das linguagens de programação C/C++ que oferece uma poderosa ferramenta computacional paralela usando placa de vídeo Nvidia. No entanto, há outros modelos de programação com placas gráficas, entre eles o OpenCL³ e o OpenACC⁴, OpenMP⁵.

Com o avanço dessa tecnologia, os grandes centros de pesquisa passaram a incluir as placas gráficas nos seus supercomputadores, fazendo-se necessário, com isso, a inclusão desta tecnologia na matriz curricular dos cursos de graduação em Computação, mas especificamente, no conteúdo da Programação Paralela e Distribuída.

Porém, como já citado em [Bachiega et al. 2017], percebe-se que o uso de placas gráficas como uma ferramenta do ensino da Programação Paralela ainda não é uma realidade na maioria dos cursos de graduação em Computação, fazendo-se, dessa forma, urgente e necessário a inclusão deste conteúdo nas disciplinas de Programação Paralela e Distribuída [Junior et. al. 2021].

4. Mapeamento Sistemático da Literatura

Um Mapeamento Sistemático da Literatura, ou simplesmente MSL, é um meio de identificar, avaliar e interpretar pesquisas relevantes para uma determinada questão de pesquisa, área ou fenômeno de interesse [Soares et al. 2019]. Também ajuda a identificar lacunas na área em que foi realizado, possibilitando assim uma melhor abordagem para novas atividades de pesquisa (Freitas Neto and Bertagnolli 2021).

¹ <https://developer.nvidia.com/cuda-zone>

² <https://www.nvidia.com/pt-br/>

³ <https://developer.nvidia.com/opencl>

⁴ <https://www.openacc.org/>

⁵ <https://www.openmp.org/>

Para isso, são realizados levantamentos de estudos primários, seguidos por uma seleção dos trabalhos retornados através de critérios de inclusão e exclusão previamente definidos. As questões de pesquisas, os critérios de seleção, bem como outras etapas necessárias são definidos no chamado protocolo de pesquisa. O objetivo deste MSL é apresentar um mapeamento sobre as metodologias que podem ser utilizadas no ensino da programação paralela utilizando placas gráficas, apontando as principais dificuldades e ferramentas utilizadas [Junior et al. 2021].

4.1. Questões de Pesquisa

A questão de pesquisa (QP) tem por objetivo auxiliar no processo de pesquisa e na extração dos dados. Para este trabalho foi definida a seguinte questão de pesquisa:

- **(QP): Quais foram as metodologias relatadas no processo de ensino-aprendizagem da programação paralela com placas gráficas?**

Para auxiliar na resposta da questão principal, foram definidas as seguintes questões secundárias (QS):

- **(QS.1) Em quais disciplinas (obrigatórias ou optativas) e em que nível de instrução (graduação ou pós-graduação) os conteúdos sobre placas gráficas foram ministrados?**
- **(QS.2) Quais foram as bibliotecas, ferramentas e/ou linguagens desenvolvidas ou utilizadas no processo de ensino-aprendizagem?**
- **(QS.3) Quais foram as aplicações desenvolvidas que colaboraram no processo de ensino-aprendizagem utilizando placas gráficas?**
- **(QS.4) Quais foram os instrumentos de avaliação utilizados para verificar o aprendizado dos alunos?**
- **(QS.5) Quais foram as principais dificuldades relatadas, por professores e alunos, durante o processo de ensino-aprendizagem?**

Essas questões foram definidas visando identificar o cenário em que as placas gráficas são utilizadas durante o ensino da programação paralela.

4.2. Processo de Busca e Seleção dos Estudos Primários

O processo de busca dos trabalhos foi realizado segundo a *string* de busca definida no Quadro 1. Foram considerados apenas estudos completos publicados em periódicos ou conferências, escritos em inglês.

Quadro 1 – String de Busca Utilizada.

Idioma	String
Inglês	<i>(“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”))</i>

Finalizado o processo de busca, foram aplicados os critérios de Inclusão e Exclusão para realizar a seleção dos estudos retornados. Os critérios de inclusão utilizados no trabalho podem ser visualizados no Quadro 2, enquanto os critérios de exclusão podem ser visualizados no Quadro 3.

Dos 867 trabalhos retornados, após a aplicação dos critérios de exclusão e inclusão, foram incluídos apenas 5 estudos. Na Tabela 1, são apresentadas as fontes utilizadas como base para a aplicação da *string* de busca, a quantidade de artigos retornados por base consultada, assim como, quantidade de trabalhos excluídos por cada critério e a quantidade de trabalhos adicionados, também, por cada critério.

Quadro 2 – Critérios de Inclusão

Critérios de Inclusão (CI)
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.

Quadro 3 – Critérios de Exclusão

Critérios de Exclusão (CE)
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 e/ou Google Scholar;
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, <i>keynote speeches</i> , cursos, tutoriais, <i>workshops</i> 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 com placas gráficas;
CE.6. Estudos que não estiverem inseridos no contexto do ensino da programação paralela com placas gráficas;
CE.7. Estudos que não estiverem apresentados no idioma aceito (Inglês);

A pesquisa foi realizada considerando os últimos 5 anos, pois foi considerada a Resolução Nº 5, de 16 de novembro de 2016 emitida pelo Ministério da Educação [MEC, 2016]. A exclusão de 424 estudos, na Tabela 1, identificado como “Excluídos 1ª Seleção” se deu pela leitura dos títulos e das palavras-chave, que não continham expressões ligadas à pesquisa.

3. Resultados e Discussões

Nesta seção serão apresentadas as respostas para as questões de pesquisa e questões secundárias definidas na elaboração do protocolo deste mapeamento sistemático. As respostas serão baseadas no número de identificação (ID) dos artigos incluídos, que estão listados e podem ser visualizados no Quadro 4.

Os quadros apresentados nas respostas foram organizados de forma a destacar os principais pontos levantados nas questões de pesquisa e questões secundárias, extraindo, dessa forma, os pontos mais relevantes dos itens questionados.

Tabela 1 – Fontes e Quantidade de Trabalhos por Critério.

Fontes	Estudos retornados	Remoção Duplicados	Excluídos 1ª Seleção	2ª Seleção								
				Excluídos							Incluídos	
				CE1	CE2	CE3	CE4	CE5	CE6	CE7	CI1	CI2
<i>IEEE</i>	13	0	5	0	2	0	0	3	1	0	1	1
<i>ACM</i>	365	12	206	1	55	2	43	34	9	0	1	2
<i>Compendex</i>	33	0	8	0	11	0	2	5	7	0	0	0
<i>Science Direct</i>	381	10	176	1	61	0	56	65	12	0	0	0
<i>ISI Web</i>	30	0	9	0	3	0	5	8	5	0	0	0
<i>Scopus</i>	45	0	20	0	9	0	9	4	3	0	0	0
TOTAL	867	22	424	2	141	2	115	119	37	0	2	3
				416							5	

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

As principais metodologias encontradas nos estudos incluídos foram destacadas no Quadro 5.

Foram identificadas seis metodologias, sendo que a mais utilizada foi a aprendizagem baseada em projetos, aparecendo em quatro, dos cinco estudos incluídos. Isso talvez possa ser explicado, por esta metodologia incentivar os alunos a aplicarem os conceitos em algo mais próximo da realidade, o que acaba por motivá-los.

(QS.1) Em quais disciplinas (obrigatórias ou optativas) e em que nível de formação (graduação ou pós-graduação) os conteúdos sobre placas gráficas foram ministrados?

O Quadro 6 apresenta as atividades onde os conteúdos da programação paralela foram trabalhados. Um fato relevante que pode ser observado é que dos cinco estudos relatados, apenas dois foram desenvolvidos em cursos de graduação em Computação, porém nenhum fazendo parte da matriz curricular obrigatória. Um dos estudos trabalhou o conteúdo como uma disciplina optativa e o outro, como parte do conteúdo da disciplina de Sistemas Operacionais. Isso aponta para uma carência, inclusive, na elaboração das disciplinas e conteúdo do ensino da programação paralela com placas gráficas nos cursos de graduação, nos últimos 5 anos, e isso não apenas no Brasil, mas também, nos cursos de graduação no exterior.

Essa carência de estudos é apontada, também, no trabalho de [Fenwick and Norris 2020], onde o autor relata a carência de produções acadêmicas que compartilhem experiências no ensino da Programação Paralela e Distribuída com placas gráficas, principalmente, com trabalhos práticos.

Quadro 4 – Identificação dos Estudos Incluídos

ID	Base	Referência
[EP.1]	IEEE	CHITRA, P.; GHAFOR, S. K. <i>Activity based approach for teaching parallel computing: An indian experience</i> . In: 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). [S.l.: s.n.]. p. 290–295. 2019.
[EP.2]	ACM	FENWICK, G.; NORRIS, C. <i>GPGPU Programming for CS undergraduates: Which one is superman?</i> In: Proceedings of the 2020 ACM Southeast Conference. [S.l.: s.n.]. p. 2–9. 2020.
[EP.3]	ACM	CHEN, J.; SHEN, L.; YIN, J.; ZHANG, C. <i>Parallel programming course development based on parallel computational thinking</i> . In: Proceedings of ACM Turing Celebration Conference - China. [S.l.: s.n.]. p. 103–109. 2018.
[EP.4]	ACM	CURTSINGER, C. <i>Parallelism in practice: experiences teaching concurrency and parallelism in an undergraduate os course</i> . In: Proceedings of the 2019 ACM SIGPLAN Symposium on SPLASH-E. [S.l.: s.n.]. 2019.
[EP.5]	IEEE	SHAMSI, J. A. <i>A laboratory based course on gpu programming: Methods, practices, and lessons</i> . In: IEEE.2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). [S.l.], p. 367–374. 2017.

Quadro 5 – Metodologias Utilizadas nos Estudos Incluídos

Metodologia	Descrição	ID
Aulas Tradicionais Expositivas	Os temas eram expostos e discutidos em sala de aula, tendo o professor como intermediador das discussões.	[EP.1] [EP.5]
Aprendizagem Baseada em Projetos	As discussões/debates realizados em sala resultavam em miniprojetos que eram expostos aos alunos para que pudessem solucioná-los em grupos e cada etapa das soluções eram avaliadas.	[EP.1] [EP.2] [EP.3] [EP.4]
Computação Desplugada	Atividades que eram realizadas sem o uso de computador, com o objetivo de incentivar a formação do pensamento paralelo e despertar o interesse do aluno.	[EP.1]
Programação Curta	Eram repassados problemas considerados de pequena complexidade (geram poucas linhas de códigos) e que era possível resolver e discutir no horário das aulas.	[EP.1] [EP.4]
Abordagem Analítica	O ensino foi dividido em duas partes: a primeira, sendo teórica, onde o conteúdo foi repassado de forma tradicional; e a segunda, de forma prática, onde os conceitos eram aplicados em um aprendizado baseado em problemas desafios.	[EP.2]
Sala de Aula Invertida	Os alunos tinham acesso aos conceitos de forma antecipada e durante as aulas, os mesmos eram discutidos no grupo como um mecanismo de incentivar a solucionar de problemas reais.	[EP.3]

Quadro 6 – Atividades Desenvolvidas

Atividades	Descrição	Formação	ID
Disciplina Específica	Computação Paralela e Distribuída	Pós-graduação em Ciência da Computação	[EP.1]
Disciplina Específica	Uma Introdução para GPGPU usando CUDA.	Curso de verão (alunos de graduação, pós-graduação e tecnólogo)	[EP.2]
Disciplina Específica	Programação Paralela baseada em Pensamento Computacional Paralelo.	Pós-graduação em Ciência da Computação	[EP.3]
Parte de uma Disciplina	Sistemas Operacionais	Graduação em Ciência da Computação	[EP.4]
Disciplina Optativa	Programação CUDA	Graduação em Ciência da Computação	[EP.5]

(QS.2) Quais foram as bibliotecas, ferramentas e/ou linguagens desenvolvidas ou utilizadas no processo de ensino-aprendizagem?

No Quadro 7 são apresentadas as ferramentas (linguagens, modelos, APIs, bibliotecas) que foram utilizadas durante os cursos de programação paralela. Como a finalidade deste trabalho é o uso de placas gráficas, o modelo de programação em CUDA está presente em todos os estudos, e uma referência é feita, também, para a API OpenCL. As bibliotecas utilizadas com processadores como OpenMP e MPI⁶, também são citadas, porém, como instrumentos iniciais para o ensino da construção do pensamento paralelo.

Essa construção do pensamento paralelo foi um dos pontos relevantes encontrados com este trabalho, pois muitos alunos apontaram dificuldades de aprendizagem quando se iniciou diretamente o curso pela linguagem CUDA, ou seja, não se iniciou pela paralelização com os processadores.

Quadro 7 – Ferramentas Utilizadas

Ferramentas Utilizadas	ID
Introdução aos conceitos básicos utilizando OpenMP e MPI, para depois iniciarem os estudos com CUDA.	[EP.1] [EP.3]
Foi utilizada a API OpenCL e CUDA.	[EP.2]
Utilização somente do CUDA.	[EP.4] [EP.5]

(QS.3) Quais foram as aplicações ou problemas solucionados que colaboraram no processo de aprendizagem utilizando placas gráficas?

Percebe-se, pelo Quadro 8, que os problemas/aplicações utilizados durante o processo de ensino-aprendizagem com placas gráficas são os mais variados possíveis, tendo aplicações em robótica, Inteligência Artificial, Engenharias, Segurança da Informação, Bioinformática, Redes de Computadores, entre outras. Isso demonstra que o uso da programação paralela, de uma forma geral, é uma realidade que vem sendo aplicada com sucesso, nas mais variadas áreas do conhecimento.

(QS.4) Quais foram os instrumentos de avaliação utilizados para verificar o aprendizado dos alunos?

⁶ <https://www.open-mpi.org/>

No Quadro 9 são apresentados os principais instrumentos utilizados para avaliar o aprendizado dos alunos. Foram aplicados desde questionários ao final de cada conteúdo, ou ao final do curso, como também pequenos códigos de rápida implementação e resolução, como uma forma de incentivar a construção do pensamento paralelo. Também foram desenvolvidas atividades práticas que eram apresentadas em sala como uma forma de incentivar a discussão sobre o assunto.

Quadro 8 – Aplicações Utilizadas nas Aulas.

Aplicações Utilizadas	ID
Simulação das correntes nos oceanos, filtragem de ruído de imagens, aplicações do algoritmo Gauss-Seidel para sistemas lineares, aplicações de classificação ímpar-par.	[EP.1]
Identificação de imagens através da comparação.	[EP.2]
Análise de dados, <i>Big Data</i> , aprendizagem profunda, reconhecimento de imagens, extração de características, detecção de objetos, rede neural, informação biológica, dinâmica dos fluidos, dinâmica de estrutura, simulação de aplicação numérica, previsão do tempo, arquitetura de sistemas operacionais de robôs, sistema de arquivos, processamento de sinal, interação humano computador, sistemas de alta precisão.	[EP.3]
Quebra de senhas, avaliações em estruturas de comunicação e sincronização por meio de GPUs.	[EP.4]
Redes de computadores, aprendizado de máquinas, processamento de texto, segurança.	[EP.5]

Quadro 9 – Instrumentos de Avaliação Utilizados

Instrumento Avaliativo	Descrição	ID
Questionários	Tinha como objetivo verificar o entendimento dos alunos sobre os tópicos ministrados.	[EP.1] [EP.5]
Códigos Curtos	Eram repassados códigos de fácil entendimento e que era solicitado ao aluno informar qual seria a saída do código.	[EP.1]
Atividade Contínua Prática	Um único problema era definido para ser solucionado, por etapas, durante o curso.	[EP.2]
Apresentações e Relatórios Experimentais	Eram definidos projetos que eram desenvolvidos com discussão em grupo e as etapas eram apresentadas, com a elaboração de relatórios experimentais.	[EP.3] [EP.4] [EP.5]

(QS.5) Quais foram as principais dificuldades relatadas, por professores e alunos, durante o processo de ensino-aprendizagem?

As principais dificuldades relatadas por professores e alunos podem ser visualizadas no Quadro 10. Percebe-se que a ausência de uma formação inicial e contínua na graduação, sobre o pensamento paralelo, acaba sendo uma das principais dificuldades apontadas pelos alunos. Uma outra dificuldade apontada pelos alunos e que merece destaque, é o fato do trabalho [EP.2] iniciar a construção do pensamento paralelo já com o modelo de programação CUDA, o que contribuiu para aumentar as dificuldades de entendimento dos alunos.

Os trabalhos [EP.3], [EP.4] e [EP.5] não são relatados no Quadro 8, já que os autores não apontaram ou relataram dificuldades encontradas durante o processo de ensino-aprendizagem. Fato relevante é que na avaliação realizada no trabalho [EP.5] não houve nenhum *feedback* negativo por parte dos alunos que participaram do curso.

Quadro 10 – Principais dificuldades relatadas

Público	Dificuldades Relatadas	ID
Alunos	Dificuldade de compreensão do pensamento paralelo por parte dos alunos, por não terem tido na graduação, disciplinas que trabalhassem com a programação paralela.	[EP.1] [EP.2]
Professores	Dificuldade de encontrar trabalhos na literatura que compartilhem experiências sobre o ensino da programação paralela com placas gráficas.	[EP.2]
Alunos	Maior complexidade na construção de códigos paralelos, assim como, o contato inicial do pensamento paralelo ter sido realizado através do CUDA.	[EP.2]

4. Conclusão

A computação paralela apesar de estar cada vez mais presente nos grandes centros de pesquisa, parece que ainda precisa ser descoberta, ou efetivamente implantada, nas instituições de ensino que oferecem curso de graduação em Computação. E essa descoberta se torna ainda mais urgente quando se relaciona o ensino da programação paralela com o uso de placas gráficas. E isso não apenas no Brasil, mas também, em boa parte do mundo [Fenwick and Norris 2020].

O uso de placas gráficas para o ensino da programação paralela se tornou um desafio para professores e instituições, isso por conta do tipo de arquitetura utilizada e pela complexidade do modelo de programação, que na maioria dos casos se utiliza o CUDA, tornando o ensino do pensamento paralelo ainda mais difícil de ser compreendido. Para contornar essas dificuldades é necessário encontrar abordagens didáticas que fixem a atenção dos alunos, despertando o interesse e preparando novos programadores para a área. E essas abordagens perpassam por estudos de metodologias que possam construir o pensamento paralelo de forma contínua e prática, fazendo que com possam observar a aplicação desses conceitos na solução de problemas da vida cotidiana.

Como trabalhos futuros, pretende-se desenvolver e aplicar metodologias que possam abordar e transmitir esses conceitos de forma pedagógica e, até porque não dizer, lúdica, com o objetivo de despertar o interesse dos alunos pela programação paralela utilizando as placas gráficas.

5. Referências

ACM and IEEE-CS (2020). *Computing Curricula: Paradigms for Global Computing Education*.

Bachiega, N. G.; Souza, P. S. L.; Bruschi, S. M.; Souza, S. R. S. (2017). Mapeamento Sistemático do Ensino Teórico e Prático de Programação Paralela In: *Workshops do Congresso Brasileiro de Informática na Educação (CBIE)*. pages. 1089-1098. Recife – PE.

Bachiega, N.; Souza, P. S. L.; Bruschi, S. M.; Souza, S. R. S. (2018). Um Panorama do Ensino de Programação Paralela e Distribuída em Universidade Brasileiras. In: *Workshops do Congresso Brasileiro de Informática na Educação (CBIE)*. pages. 480-489. Fortaleza – CE.

Curtsinger, C. (2019). Parallelism in practice: experiences teaching concurrency and parallelism in an undergraduate os course. In: *Proceedings of the ACM SIGPLAN Symposium on SPLASH-E*. [S.l.: s.n.]. pages. 1–6. 2019.

Fenwick, G.; Norris, C. (2020) GPGPU Programming for CS Undergraduates: Which one is Superman? *ACM Southeast Conference – ACMSE*. pages. 2-9. Tampa – FL (USA).

Freitas Neto, J. J.; Bertagnolli, S. C. (2021). Robótica Educacional e Formação de Professores: In.: Uma Revisão Sistemática da Literatura. In *Revista Novas Tecnologias na Educação*. Volume. 19. Nº 1.

Galvão, M. C. B.; Ricarte, I. L. M. (2019). Revisão Sistemática da Literatura: Conceituação, Produção e Publicação. In: *Logeion: Filosofia da Informação*. Volume. 6. Nº 1. pages. 57-73.

Gardner, W. B. (2017) *Should we be teaching parallel programming?* In: *Proceedings of the 22th Western Canadian Conference on Computing Education*.

Junior, R. B. A., Farias, H., Saavedra, M., and de Souza Araújo, J. (2021). Metodologias de ensino na programação paralela com placas gráficas: Uma revisão sistemática da literatura. In *Anais do X Congresso Brasileiro de Informática na Educação (CBIE)*, pages 1243-1252. Evento On-line.

Lima, A. C.; Bailo, D.; Carvalho, T.; Rodrigues, J. F. R.; Acosta, P. R.; Aquino, W. M. (2018) Uma Oficina para Ensino de Algoritmos Paralelos por Meio de Computação Desplugada. In: *Congresso Brasileiro de Informática na Educação – CBIE*. pages. 619-628. Fortaleza – CE.

MEC. Resolução nº 5, de 16 de novembro de 2016. (2016). In: *Diário Oficial da União (DOU)*. Edição 220, Seção 1, page. 22.

Saeed, I. (2020). *Why CS Departments Should Consider Offering CUDA as a Standalone Course*. In *Journal of Computing Sciences in Colleges*. pages.51-58.

Silva, T. R. S.; Barros, I. S.; Sousa, L. K. S.; Sá, A. L. D.; Silva, A. F. M.; Araujo, M. C. S.; Aranha, E. H. S. (2021) Um Mapeamento Sistemático sobre o Ensino e Aprendizagem de Programação. In.: Uma Revisão Sistemática da Literatura. In *Revista Novas Tecnologias na Educação*. Volume. 19. Nº 1.

Silva filho, F. R.; Coutinho, E. F. (2021) Uma Análise Qualitativa sobre as Disciplinas de Fundamentos de Programação e Estrutura de Dados com Grounded Theory. In *Revista Novas Tecnologias na Educação*. Volume. 19. Nº 1.

Soares, F.; Nobre, C.; Freitas, H. (2019). *Parallel Programming in Computing Undergraduate Courses: A Systematic Mapping of the Literature*. In *IEEE Latin America Transactions*. Volume. 17. Nº 8. pages. 1371-1381.

Zorzo, A. F.; Nunes, D.; Matos, E.; Steinmacher, I.; Araujo, R. M. de; Correia, R.; Martins, S. (2017). Referenciais de Formação para os Cursos de Graduação em Computação. In *Sociedade Brasileira de Computação (SBC)*.