

Um Mapeamento da Literatura sobre o Uso de Ferramentas para Auxiliar no Processo de Ensino e Aprendizagem da Programação Paralela

Marcos Saavedra¹, Thaís Quaresma¹, Marcus Oliveira¹, Josivaldo Araújo¹

¹Instituto de Ciências Exatas e Naturais – Universidade Federal do Pará (UFPA)
Rua Augusto Corrêa nº 01 - 66.075-110 - Belém - PA - Brasil

{marcosaavedraa3, thaismquaresma, marq938}@gmail.com,
josivaldo@ufpa.br

Abstract. *The objective of this work is to report the results found from a Systematic Mapping of Literature (MSL) on tools and/or online environments that can serve as support in the teaching and learning process of Parallel Programming. Four international databases were used and thirteen papers considered relevant in this process were identified. Among the main points highlighted are the main concepts of Parallel Programming, libraries, languages, learning methodologies, description of tools and/or environments found, as well as the evaluation of teachers and students in the processes used.*

Resumo. *O objetivo deste trabalho é relatar os resultados encontrados de um Mapeamento Sistemático da Literatura (MSL) sobre ferramentas e/ou ambientes on-line que podem servir de apoio no processo de ensino e aprendizagem da Programação Paralela. Foram utilizadas quatro bases internacionais e identificados treze trabalhos considerados relevantes neste processo. Entre os principais pontos destacados estão os principais conceitos de Programação Paralela, bibliotecas, linguagens, metodologias de aprendizagem, descrição das ferramentas e/ou ambientes encontrados, bem como a avaliação de professores e alunos nos processos utilizados.*

1. Introdução

Os sistemas *multicore* e *manycore* ganharam uma grande visibilidade nos últimos anos devido à necessidade de se processar uma quantidade cada vez maior de dados em áreas como a Inteligência Artificial e Ciência de Dados, por exemplo. Como consequência, houve uma demanda por profissionais que pudessem atuar em projetos que melhorassem o desempenho das aplicações, reduzissem o tempo de processamento e otimizassem o uso dos recursos do sistema [Perez et al. 2023].

Com o propósito de promover uma formação específica e uma melhor qualificação, muitos cursos de graduação em Computação inseriram em suas matrizes curriculares o ensino da Programação Paralela. Para isso, diversas diretrizes foram elaboradas com o objetivo de fornecer uma base solidificada a partir dos conceitos da Computação de Alto Desempenho, proporcionando o desenvolvimento de competências que permitam, não apenas a construção de códigos paralelos, mas também um ganho de desempenho e uma melhor otimização das aplicações e dos sistemas [Galante 2023].

Contudo, para que essas habilidades sejam desenvolvidas, ainda há a necessidade de se superar alguns obstáculos no processo de ensino e aprendizagem da Programação Paralela, como: o acesso às arquiteturas verdadeiramente paralelas, o uso de bibliotecas próprias e a efetiva compreensão dos conceitos paralelos. Desses obstáculos, o mais citado na literatura é a falta de laboratórios específicos, principalmente, àqueles que possam disponibilizar o uso das placas gráficas, como apontou [Jr. et al. 2022]. Com isso, muitos professores acabam limitando o ensino apenas ao uso dos processadores (ou *cores*). Outra forma de contornar esses obstáculos é através do uso de diversas metodologias e/ou ferramentas que acabam sendo utilizadas com o objetivo de favorecer e promover a construção do ensino da Programação Paralela [Dias and Araújo 2024].

O objetivo deste trabalho é apresentar um Mapeamento Sistemático da Literatura (MSL) sobre o uso de ferramentas que possam auxiliar no processo de ensino e aprendizagem da Programação Paralela. Para isso, esse mapeamento apontará as principais plataformas e/ou ferramentas desenvolvidas, quais são os principais conceitos abordados, as principais metodologias utilizadas e a avaliação de professores e alunos sobre essas ferramentas e estratégias pedagógicas utilizadas, destacando as principais vantagens e desvantagens.

2. Trabalhos Relacionados

[Diogo et al. 2023] realiza uma revisão da literatura a respeito de uma investigação sobre os efeitos do uso de metodologias ativas no ensino de programação. Foram encontrados 24 artigos que relatam diversas metodologias sendo aplicadas em disciplinas do ensino básico e da graduação. Sendo a metodologia baseada em problemas a mais utilizada.

O trabalho de [Jr. et al. 2022] apresenta uma revisão sobre metodologias e ferramentas utilizadas no ensino da Programação Paralela e que utilizam placas gráficas. A pesquisa obteve cinco resultados que apontaram o uso de metodologias tradicionais (aulas expositivas) e ativas (computação desplugada, aprendizado baseado em projetos e salas de aula invertidas).

[Fernandes et al. 2023] realizou um estudo em bases internacionais sobre o uso de ambientes digitais utilizados no treinamento de profissionais da educação. Esta busca foi realizada em bases internacionais, onde foram identificados 41 artigos que apontam o uso de ferramentas digitais que atuam no contexto da pesquisa.

Em comparação aos trabalhos anteriores, esta pesquisa realizou um MSL sobre o uso de ferramentas de ensino que os estudantes possam acessar via web, *on-line* ou projetos que estejam disponíveis em repositório aberto, que tenham o processo de acesso e instalação facilitado e conteúdos referentes à Programação Paralela.

3. Mapeamento Sistemático da Literatura (MSL)

Um MSL é uma abordagem utilizada para investigação de um determinado tema proposto a partir de uma classificação primária de estudos [Hyury et al. 2024]. Esta metodologia é reconhecida por classificar objetos de pesquisa em uma área abrangente a que se pretende analisar, sendo considerada útil em análise de bases onde retornam muitas publicações [Ferreira and Bernardini 2024].

As fases do MSL são definidas por meio de um protocolo, previamente elaborado. Este processo é considerado importante, pois ajuda na condução da coleta dos dados e a

realizar uma análise dos retornados [Hyury et al. 2024]. Uma das fases desse protocolo é a definição da Questão de Pesquisa (QP) e das Questões Secundárias (QS). Para este MSL foram estabelecidos os questionamentos referentes ao processo de avaliação, apontando uma Questão Principal (QP) e as Questões Secundárias (QS) para auxiliar na avaliação dos estudos:

- [QP]: **Quais são as plataformas ou ferramentas desenvolvidas que contribuem no processo de ensino e aprendizagem da Programação Paralela nos cursos de Computação?**
- [QS.1]: Quais foram as linguagens de programação utilizadas para o ensino da Programação Paralela?
- [QS.2]: Quais são os *frameworks* ou bibliotecas paralelas utilizadas para desenvolver o ensino de aplicações paralelas?
- [QS.3]: Quais foram os conceitos de Programação Paralela abordados nas plataformas ou ferramentas utilizadas?
- [QS.4]: Quais são as estratégias pedagógicas utilizadas no processo de ensino e aprendizagem da Programação Paralela?
- [QS.5]: Como são disponibilizados os recursos pelas ferramentas ou plataformas (*on-line*, instalação/configuração local, gratuitos, pagos)?
- [QS.6]: Qual a avaliação de professores e alunos com relação às metodologias e/ou recursos educacionais utilizados no processo de ensino e aprendizagem?

A condução do MSL requer a identificação dos critérios de aceitação. Para um trabalho ser considerado aceito ou rejeitado, ele deve ser avaliado a partir dos critérios de inclusão ou exclusão definidos nesta etapa de planejamento. Os critérios de aceitação podem ser visualizados na Tabela 1 e os critérios de rejeição podem ser visualizados na Tabela 2.

ID	Critério de Inclusão (CI)
CI.1.	Estudos que apresentem abordagens, no contexto de plataformas ou ferramentas computacionais, que dão apoio às atividades de ensino e aprendizagem da Programação Paralela;
CI.2.	Estudos que apresentem relatos de experiência, com exemplos de aplicação, descrição de experimentos ou casos reais de uso de abordagens, no contexto da Programação Paralela, para apoio às atividades de ensino e aprendizagem.

Tabela 1. Critérios de Inclusão

Foi utilizado o método PICOC¹ para a formação da *String de busca*. Neste método, são descritos os elementos fundamentais para a realização do mapeamento sistemático [Saavedra et al. 2024]. A questão principal da pesquisa aponta de forma explícita as fases do método PICOC: *Quais são as (P) plataformas ou ferramentas desenvolvidas que contribuem no (O) processo de ensino e aprendizagem da (I) Programação Paralela nos (C) cursos de Computação?*

A partir dessa estratégia, é definida uma *string* de busca que será utilizada nas bases de repositórios de artigos. A Tabela 3 destaca a *string* usada neste trabalho. Vale ressaltar que a mesma foi adaptada às regras de uso de cada repositório.

¹População, Intervenção, Comparação, Resultado e Contexto

ID	Critério 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 enquadrados como resumos, <i>keynote</i> , <i>speeches</i> , cursos, tutoriais, <i>workshops</i> e afins;
CE.3.	Trabalhos que estejam fora do período estipulado para a pesquisa;
CE.4.	Estudos que não estejam escritos em português ou inglês;
CE.5.	Estudos Secundários.
CE.6.	Estudos repetidos (em mais de uma fonte de busca) terão apenas sua primeira ocorrência considerada.

Tabela 2. Critérios de Exclusão

Idioma	<i>String</i>
Português	(“Plataforma”OR “Ferramentas”) AND (“Programação Paralela”) AND (“Ensino”OR “Aprendizagem”) AND (“Ciência da Computação”OR “Computação”)
Inglês	(“Platforms”OR “Tools”OR “Environments”) AND (“Parallel Programming”) AND (“Teaching”OR “Learning”) AND (“Computer Science”OR “Computing”)

Tabela 3. Strings de busca utilizadas nos repositórios de artigos

Após ser realizada a validação, foi aplicada a *string* nas bases de quatro repositórios para a seleção dos trabalho: *El Compendex*², *IEEE Digital Library*³, *Science Direct*⁴ e *Scopus*⁵.

O período definido para a pesquisa dos estudos nos repositórios foi de janeiro de 2013, ano de definição, pela ACM e IEEE, da matriz curricular de Programação Paralela para o curso de Ciência da Computação, até maio de 2025. Para auxiliar na organização dos trabalhos, foi utilizada a plataforma Parsifal⁶.

A seleção dos artigos foi realizada por uma equipe composta por dois graduandos e um pós-graduando. A coordenação foi realizada por um professor que foi o responsável pela pesquisa. O processo de seleção foi realizado em três fases. Na primeira fase, cada participante, de posse do total de artigos retornados pelas bases utilizadas, leu apenas o título e o resumo dos artigos. Na segunda fase, os artigos selecionados na primeira fase (de forma individual), tiveram lidos o resumo, a introdução e a conclusão dos artigos. Sendo esse resultado comparado entre os integrantes da equipe. Na terceira fase, foram lidos os trabalhos completos. Em todas as fases foram aplicados os critérios de inclusão e exclusão, sendo que na primeira, os critérios foram adotados de forma individual (por cada participante) e nas fases seguintes em conjunto (os critérios eram discutidos entre os participantes no MSL).

²<https://www.engineeringvillage.com/search/quick.url>

³<https://ieeexplore.ieee.org/Xplore/home.jsp>

⁴<https://www.sciencedirect.com/>

⁵<https://www.scopus.com/>

⁶<https://parsif.al/>

Bases	Artigos Retornados	Rejeitados						Aceitos	
		CE.1	CE.2	CE.3	CE.4	CE.5	CE.6	CI.1	CI.2
Compendex	171	4	12	0	3	57	93	2	0
IEEE	104	1	5	23	0	57	18	0	0
ScienceDirect	1699	1	188	1	0	1452	53	4	0
Scopus	1667	7	7	0	0	1525	121	6	1
Total	3641	13	212	24	3	3091	285	12	1

Tabela 4. Resultado da seleção dos estudos

4. Resultados

Este MSL retornou um total de 3641 estudos, 13 desses estudos atenderam aos Critérios de Inclusão e 3628 foram rejeitados. O alto número de trabalhos excluídos pode ser explicado por serem trabalhos secundários, o que atende ao Critério de Exclusão (CE.5). O resultado da aplicação dos Critérios de Exclusão (CE) e Inclusão (CI) pode ser visualizado na Tabela 5.

As respostas das Questões Principal e Secundárias estão definidas pelo identificador (ID) dos Estudos Primários (EP), da Tabela 4. Os trabalhos aqui relacionados atenderam aos Critérios de Inclusão (CI) definidos na Tabela 1.

QP: Quais são as plataformas ou ferramentas desenvolvidas que contribuem no processo de ensino e aprendizagem da Programação Paralela nos cursos de Computação?

Pode-se perceber que há diferenças significativas entre as ferramentas e os ambientes, desde sistemas baseados em blocos, passando por ferramentas que avaliam os códigos de forma automática, chegando até tutores inteligentes. Cada um aplicando as suas metodologias, personalizando e conduzindo o aprendizado da Programação Paralela.

Em [EP.1] é apresentado o "SNAP!" um *framework* visual baseado em blocos, projetado para ensinar os conceitos de Programação Paralela de forma lúdica, possibilitando aos estudantes visualizarem o funcionamento dos algoritmos desenvolvidos. Em [EP.2] é abordada a ferramenta web "SAUCE", que avalia automaticamente os códigos paralelos. Essa ferramenta é integrada às bibliotecas do MPI e do OpenMP, facilitando o *feedback* aos estudantes e promovendo uma melhor compreensão dos conteúdos lecionados. Em [EP.3] é apresentado o "ONRAMP" que é um portal web com diversos recursos educacionais, como tutoriais e exercícios voltados ao ensino de Computação Paralela e Distribuída. Já o estudo [EP.4] descreve o "COMP 322" que é um curso de Programação Paralela que utiliza duas ferramentas que auxiliam no aprendizado da Programação Paralela, são elas a biblioteca HJlib e a ferramenta Habanero Autograder.

O estudo [EP.5] destaca a plataforma "EDUWRENCH" que é uma plataforma educacional com uma trilha de ensino sobre a Computação Paralela e Distribuída. Os assuntos abordados podem ser modelados a partir da ferramenta WRENCH que é um *framework* de simulação em C++/Python que permite modelar ambientes distribuídos,

ID	Base	Referência
EP.1	<i>El Compendex</i>	<i>Parallel programming with pictures is a Snap!</i>
EP.2	<i>El Compendex</i>	<i>SAUCE: A web-based automated assessment tool for teaching parallel programming</i>
EP.3	<i>Science Direct</i>	<i>OnRamp: A web-portal for teaching parallel and distributed computing</i>
EP.4	<i>Science Direct</i>	<i>Pedagogy and tools for teaching parallel computing at the sophomore undergraduate level.</i>
EP.5	<i>Science Direct</i>	<i>Teaching parallel and distributed computing concepts in simulation with WRENCH</i>
EP.6	<i>Science Direct</i>	<i>Using Cognitive Computing for Learning Parallel Programming: An IBM Watson Solution</i>
EP.7	<i>Scopus</i>	<i>Cloud E-learning and benchmarking platform for the parallel and distributed computing course</i>
EP.8	<i>Scopus</i>	<i>Integrating Interactive Performance Analysis in Jupyter Notebooks for Parallel Programming Education</i>
EP.9	<i>Scopus</i>	<i>Interactive Textbooks for Parallel and Distributed Computing Across the Undergraduate CS Curriculum</i>
EP.10	<i>Scopus</i>	<i>Let's HPC: A web-based platform to aid parallel, distributed and high performance computing education</i>
EP.11	<i>Scopus</i>	<i>On-line service for teaching parallel programming</i>
EP.12	<i>Scopus</i>	<i>P2RUTOR: A Programming Tutor for Parallel Programming</i>
EP.13	<i>Scopus</i>	<i>Tareador: A tool to unveil parallelization strategies at undergraduate level</i>

Tabela 5. Estudos Incluídos no Mapeamento Sistemático da Literatura

ajudando estudantes a compreender os desafios sem necessidade de uma infraestrutura física. Em [EP.6] apresenta o ”IBM Watson Solution” um sistema que apresenta o conceito de Computação Cognitiva, que auxilia no aprendizado do paralelismo, oferecendo explicações personalizadas e correção automatizada. No trabalho [EP.7] é apresentado o ”Cloud E-learning Platform”, um ambiente em nuvem para ensino e avaliação de sistemas voltados para Programação Paralela e Distribuída, com suporte a tecnologias como *Hadoop* e *Spark*, que são ferramentas utilizadas para armazenamento e processamento de *Bigdata*. Já o estudo [EP.8] utiliza o ”Jupyter Notebook” e, também, foi desenvolvida uma extensão para integrar a análise de desempenho paralelo em um ambiente onde o usuário possa interagir, facilitando o aprendizado em aulas práticas.

O estudo [EP.9] apresenta o ”LEARNPDC” um *e-book on-line* com os conceitos da Computação de Alto Desempenho. A proposta deste trabalho é fornecer material com conceitos importantes para estudantes iniciantes e intermediários. O trabalho [EP.10] ”Let's HPC” é uma plataforma web de ensino de Computação de Alto Desempenho com tutoriais, exercícios e simulações utilizando diversas ferramentas de Programação Paralela. [EP.11] utiliza o ”ZAWODYWEB” que é um serviço remoto usado em aulas práticas de programação paralela, sendo acessível via navegador. O [EP.12] P2RUTOR é um tutor inteligente que auxilia no aprendizado da Programação Paralela, fornecendo *feedback*

automatizado e adaptativo aos estudantes. E [EP.13] apresenta o "TAREADOR", uma ferramenta para visualizar estratégias de paralelismo em código OpenMP e que ajuda os estudantes a identificar melhorias de otimização em seus algoritmos.

[QS.1] Quais foram as linguagens de programação utilizadas para o ensino da Programação Paralela?

As linguagens de programação mais utilizadas podem ser observadas na Tabela 6. Observa-se que as principais linguagens de Programação Paralela continuam a ser utilizadas no processo de aprendizagem, ou seja, C, C++ e Fortran. Destaque para as linguagens Java e Python. Isso pode ser explicado, pois o MPI, possui suporte para a linguagem de programação Python. Já a linguagem Java é utilizada para introduzir conceitos, principalmente, relacionados às *Threads*, que são a base da biblioteca OpenMP, por exemplo.

ID	Linguagem de Programação
[EP.1] [EP.3] [EP.8]	C/C++
[EP.2] [EP.7]	C++
[EP.6] [EP.13]	C/C++ e Fortran
[EP.4]	Java
[EP.10] [EP.12]	C/C++ e Python
[EP.9]	C/C++, Python e Java
[EP.11]	Java e C
[EP.5]	Não define linguagem

Tabela 6. Linguagens utilizadas para Ensino de Programação Paralela

[QS.2] Quais são os frameworks ou bibliotecas paralelas utilizadas para desenvolver o ensino de aplicações paralelas?

Os frameworks e as bibliotecas citados no MSL utilizados para o ensino da Programação Paralela podem ser vistos na Tabela 7. Nela são destaque o OpenMP, o MPI, o OpenACC e CUDA. Mais uma vez, frameworks baseados em Java aparecem, pois são muito utilizados, como já citados, para exemplificar conceitos e construir códigos usando *Threads*.

ID	Bibliotecas/Frameworks
[EP.1] [EP.6]	OpenMP
[EP.2] [EP.3][EP.13]	OpenMP, MPI e CUDA
[EP.5] [EP.12]	MPI
[EP.4]	(Hjib) Java
[EP.7] [EP.8] [EP.10]	OpenMP e MPI
[EP.9]	OpenMP, MPI, OpenACC e CUDA
[EP.11]	OpenMP, MPI e PJC (<i>Parallel Computing in Java</i>)

Tabela 7. Frameworks e bibliotecas utilizados para Ensino de Programação Paralela

[QS.3] Quais foram os conceitos de Programação Paralela abordados nas plataformas ou ferramentas utilizadas?

No estudo [EP.1] foram aplicados diversos conceitos referentes à Programação Paralela, sendo criadas funções dentro da ferramenta que simulam a paralelização. Uma dessas funções é o bloco *parallelMap*, que aplica uma função a cada elemento de uma lista em paralelo. Outro bloco criado é *parallelForEach* que executa de forma paralela os elementos de uma lista. Outros conceitos são abordados, como otimizar o processamento de grandes volumes de dados, mapeando e reduzindo o tempo de execução com paradigmas paralelos, e o conceito de concorrência em multitarefas. Já o estudo [EP.2] utiliza paralelização em alguns algoritmos (multiplicação de matrizes, equação de *Poisson*) e alguns conceitos como o uso de *Threads* para auxiliar no processamento, comunicação assíncrona e síncrona (utilizando MPI), acesso à memória compartilhada, condição de corrida, uso de memória compartilhada em CUDA e como melhorar os resultados de processamento. Em [EP.3] é definido o uso de diversos conceitos como decomposição de domínio (como decompor tarefas e dados), escalabilidade, Lei de Amdahl, arquiteturas paralelas de memória compartilhada, memória distribuída, *threads* e processos, sincronização e balanceamento de carga. Já em [EP.4] foram usados conceitos de modelos de paralelismo (*Fork-Join*, SIMD⁷, SPMD⁸), sincronização, concorrência e programação distribuída.

Já em [EP.5] noções de paralelização, avaliação de *speedup* e eficiência foram os principais assuntos abordados. Em [EP.6] foram relatados conceitos a respeito das diretrizes de compilação do OpenMP, sua utilização em arquiteturas de memória compartilhada e os erros mais comuns cometidos por programadores. Em [EP.7] foi realizada uma comparação entre implementações sequenciais e paralelas, análise de desempenho com os cálculos de *speedup* e eficiência, o uso de *threads* e processos. Já em [EP.8] foram usados os conceitos de sincronização, escalonamento, comunicação entre processos e análise de desempenho. Em [EP.9] foram utilizados diversos conceitos referentes à Programação Paralela, como os padrões de paralelismo, modelos de arquiteturas paralelas, as métricas de desempenho e diversas aplicações práticas.

Os conceitos desenvolvidos em [EP.10] foram os modelos de arquitetura (memória compartilhada e distribuída) e análise de desempenho (*speedup*, eficiência e lei de Amdahl). Em [EP.11] foram definidos os assuntos de memória compartilhada, passagem de mensagem através do MPI, sincronização, comunicação, operações assíncronas, escalonamento e execução em ambientes distribuídos. Em [EP.12] foram apresentados os conceitos de comunicação ponto a ponto (*peer-to-peer*), decomposição de domínio, escalonamento de processos em várias unidades de processamento e o gerenciamento de recursos compartilhados. Em [EP.13], utilizou-se os assuntos de decomposição de tarefas, dependência entre as tarefas (análise de dependência de dados e como tratar essas dependências), balanceamento de carga (identificando o desequilíbrio de carga e solucionando o problema), sincronização, escalabilidade e os modelos de Programação Paralela (OpenMP, MPI e em GPU).

[QS.4] Quais são as estratégias pedagógicas utilizadas no processo de ensino e aprendizagem da Programação Paralela?

O estudo [EP.1] aponta o uso de estratégias visuais, abstraindo conceitos complexos a partir de exemplos práticos e o aprendizado por meio de projetos. Em [EP.2], com a

⁷Single Instruction Multiple Data

⁸Single Program Multiple Data

ajuda da plataforma SAUCE, os estudantes puderam ter um *feedback* imediato. Uma outra proposta usada foi a resolução de exercícios práticos, com o fornecimento de trechos de códigos e orientações para que os estudantes pudessem concluir o desenvolvimento. Comparação dos resultados e detecção de plágio também foram utilizados. No estudo [EP.3] os professores abordaram os conceitos previamente e os estudantes puderam configurar alguns parâmetros de execução (como o número de *threads*, por exemplo). O nível dos conceitos foi aumentando de acordo com o avanço da disciplina e os alunos tiveram um aprendizado progressivo. Já em [EP.4] foram utilizadas aulas convencionais e atividades com Metodologias Ativas, dentre elas, sala de aula parcialmente invertida (os alunos preenchem uma planilha de onde revisam os conteúdos no início da aula, assistem a vídeos *on-line*, respondem *quizzes* e atividades a partir de vídeos apresentados), também possuem sessões práticas de laboratório.

No estudo [EP.5] foram usadas estratégias baseadas em simulação de problemas, analisando cenários e resolução de questões por meio de perguntas guiadas. No estudo [EP.6] a abordagem utilizada foi a interação com a ferramenta da IBM, utilizando a linguagem natural, por meio da realização de perguntas referentes ao OpenMP. Já em [EP.7] as técnicas usadas foram baseadas na exposição de exemplos e execuções de atividades práticas. Em [EP.8] a proposta adotou as técnicas de atividades práticas por meio de uma ferramenta acessível e o uso de materiais assíncronos (vídeos e materiais de consulta). Já em [EP.9] o trabalho adotou a abordagem conceitual, com uso de conteúdo simples até os mais complexos, analogias dos conceitos e atividades desplugadas.

No trabalho [EP.10] as estratégias pedagógicas utilizadas foram aprendizado baseadas em projetos, onde os alunos desenvolveram algoritmos paralelos e realizaram a análise de desempenho. Avaliação contínua, onde a combinação de atividades teóricas, práticas e apresentações orais foram adotadas para avaliar o desempenho da turma. Em [EP.11], para o ensino da Programação Paralela, foi adotada a estratégia de aprendizado prático, onde os alunos desenvolveram soluções para problemas reais, podendo utilizar uma infraestrutura paralela. Em [EP.12], a proposta apresentada utilizou estratégias de desenvolvimento de atividades práticas, oferecendo suporte durante o processo de desenvolvimento dos alunos. E em [EP.13], foram utilizadas o uso de metodologias baseadas em projetos, com aulas práticas dos conceitos ensinados e realizando uma transição gradual da programação sequencial para a paralela. As atividades foram realizadas em laboratório guiado.

[QS.5] Como são disponibilizados os recursos pelas ferramentas ou plataformas (*on-line*, instalação/configuração local, gratuitos, pagos)?

No trabalho [EP.1] a ferramenta utilizada está disponível de forma *on-line*⁹ gratuita, podendo ser acessada pelo navegador e sem a necessidade de instalação. Em [EP.2], a ferramenta SAUCE é uma plataforma web, aberta e está disponível em um repositório no Github¹⁰. É necessário um ambiente Unix com suporte a Docker para a instalação. Ja em [EP.3] o portal é disponibilizado de forma gratuita no Github¹¹, sem a necessidade de instalação. Em [EP.4] a ferramenta *Habanero Autograder* está disponível em repo-

⁹<https://snap.berkeley.edu/>

¹⁰<https://github.com/moschlar/SAUCE>.

¹¹<https://github.com/OnRampOrg/onramp>

sítorio no Github¹², com componentes podendo ser executados em um cluster SLURM para testes de desempenho.

No estudo [EP.5] o simulador está disponível de forma gratuita em repositórios web no Github¹³. A plataforma educacional pode ser acessada *on-line*¹⁴. Já em [EP.6] A ferramenta *IBM Watson Dialog Service* foi descontinuada e foi substituída pelo *IBM Watson Dialog Conversation Service*. Ela encontra-se disponível na *on-line*¹⁵, o documento não menciona se está disponível de forma gratuita, mas o site disponibiliza um período de teste. Já em [EP.7] o estudo aponta que a ferramenta é *on-line*, com execução em servidores na nuvem. Não foi mencionada a URL ou código-fonte do projeto. O estudo [EP.8] destaca que o *Jupyter* está disponível *on-line*¹⁶. As outras ferramentas desenvolvidas estão disponíveis em repositório aberto¹⁷. Em [EP.9] os dois *e-books* estão disponíveis de forma gratuita na internet¹⁸ e o código-fonte é livre¹⁹.

No trabalho [EP.10] os recursos estão disponíveis *on-line*²⁰ e de forma gratuita. O código-fonte está disponível no Github²¹, podendo ser configurado localmente. O projeto utilizou uma arquitetura de 5 nós e 80 núcleos. Já o estudo [EP.11], destacou o sistema *ZawodyWeb* que pode ser acessado via navegador. Ele é gratuito e de código aberto, sendo disponibilizado no github²². [EP.12] O projeto deve ser configurado em ambiente local. O código encontra-se disponível no github²³. E em [EP.13] as ferramentas do *Tareador*, *Extriae* e *Paraver* estão disponíveis para instalação local em plataformas Linux²⁴.

[QS.6] Qual a avaliação de professores e alunos com relação às metodologias e/ou recursos educacionais utilizados no processo de ensino e aprendizagem?

O [EP.1] foi aplicado em um evento de Computação para estudantes de escolas do ensino fundamental. Os professores afirmaram que houve um bom engajamento entre os participantes e pretendem aplicar o método em estudantes do ensino superior. Já em [EP.2] os professores apontam que a ferramenta auxilia o estudante, promovendo correções de maneira eficaz, reduzindo frustrações referentes ao aprendizado da Programação Paralela. Em [EP.3] os testes foram aplicados na disciplina de Arquitetura de Computadores a graduandos de Computação. Os professores apontaram que o portal abstraiu conceitos de complexidade técnica e reduziu a curva de aprendizado. Já em [EP.4] houve um *feedback* positivo com relação a alunos e professores. A ferramenta *Autograder* oferece suporte ao ensino, ajudando a corrigir os erros e facilitando o aprendizado. No estudo [EP.5] a proposta foi bem aceita, havendo um bom engajamento no uso da ferramenta e um *feedback* positivo dos estudantes.

¹²<https://github.com/agrippa/habanero-autograder>

¹³<https://github.com/henricasanova/wrench-serverless>

¹⁴<https://eduwrench.org>

¹⁵<https://www.ibm.com/watsonx>

¹⁶<https://jupyter.org/>

¹⁷<https://zenodo.org/records/10573107>

¹⁸<https://www.learnpdc.org/>

¹⁹<https://github.com/csinparallel/CSinParallel>

²⁰<http://www.letshpc.org/>

²¹<https://github.com/letshpcorg>

²²<https://github.com/faramir/ZawodyWeb>

²³<https://github.com/deepakbh21/Node-allocator>

²⁴<https://www.bsc.es/discover-bsc/organisation/scientific-structure/performance-tools>

Já em [EP.6] foram realizados testes e coletados dados de 8 participantes que consideraram a ferramenta útil para o aprendizado de Programação Paralela. Em [EP.7] a ferramenta teve uma primeira versão que era executada em um servidor local e foi reportado que os desenvolvedores tiveram problemas de concorrência, ocasionando resultados não confiáveis. A segunda versão utilizou servidores em nuvem e obteve um desempenho consistente, promovendo a escalabilidade das aplicações. Já em [EP.8] o trabalho destaca que o ambiente é de fácil uso, auxiliando no aprendizado da Programação Paralela. Como pontos negativos, a ferramenta ocasionou lentidão em alguns momentos do teste. Em [EP.9] o material poderá ser usado como apoio no ensino da Programação Paralela. O estudo [EP.10] aponta que os alunos consideraram a plataforma útil para o ensino da Programação Paralela, devido à facilidade de acesso e às ferramentas de automação disponíveis. Os instrutores destacaram que os relatórios automatizados gerados ajudaram na avaliação de desempenho da turma.

No trabalho [EP.11] os alunos conseguiram utilizar a ferramenta para resolver problemas paralelos. Eles consideraram que é uma plataforma fácil de usar, permitindo focar nos conceitos da paralelização. Em [EP.12] os tutores apontaram que houve uma redução da carga de trabalho, pois a ferramenta auxiliou no acompanhamento do progresso dos estudantes. Os estudantes, por sua vez, destacaram que, com a ferramenta, houve uma facilitação no aprendizado dos assuntos e no desenvolvimento das atividades; contudo, houve uma dependência de disponibilidade de nós (unidades de processamento) durante picos de submissão, o que pode afetar a experiência em cursos com um número muito grande de participantes. E em [EP.13] o trabalho destacou que a proposta foi validada com estudantes de graduação e alunos de mestrado. Os estudantes conseguiram realizar a análise das paralelizações e as ferramentas contribuíram com o aprendizado, tendo um impacto positivo no ensino da Programação Paralela.

5. Ameaça à Validade da Pesquisa

As possíveis ameaças que podem afetar este MSL são de qualidade descritiva e teórica. Na validade descritiva, o processo de avaliação não adotou um formulário com extração de dados quantitativos (pontuando a cada estudo) para que o trabalho fosse considerado aceito. Devido ao objetivo deste trabalho ser de apenas identificar ferramentas que possam servir de suporte ao processo de ensino e aprendizagem da Programação Paralela, os estudos aceitos responderam apenas às Questões de Pesquisa (QP) e Secundárias (QS) propostas.

A ameaça à validade teórica reside na possibilidade de falsos negativos devido à extensa quantidade de trabalhos retornados pelas bases. Com o objetivo de reduzir esta ameaça, este MSL foi conduzido por três pessoas separadamente. Após essa etapa, o resultado da seleção final foi debatido entre os pesquisadores.

6. Conclusão

Este MSL identificou e analisou ferramentas e plataformas utilizadas no processo de ensino e aprendizagem da Programação Paralela em cursos de Computação, destacando as principais metodologias utilizadas nesse processo. A pesquisa revelou que as ferramentas abordadas desempenham um papel fundamental na simplificação de conceitos complexos, no fornecimento de *feedback* e na promoção de um aprendizado prático, proporcionando

ao estudante a capacidade de interagir com as ferramentas. Além disso, a diversidade de linguagens de programação, *frameworks* e conceitos abordados demonstra a abrangência e a adaptabilidade dessas ferramentas a diferentes contextos educacionais.

Os resultados indicam que a combinação de estratégias pedagógicas e o uso de plataformas acessíveis, via web, contribuem significativamente para o engajamento dos estudantes e para a superação de desafios no ensino da Programação Paralela. No entanto, a pesquisa também aponta para a necessidade de maior investimento em infraestrutura e na formação de professores para que essas ferramentas sejam utilizadas em todo o seu potencial. Futuros trabalhos podem explorar a integração dessas ferramentas em currículos acadêmicos e a avaliação de seu impacto a longo prazo na formação dos estudantes.

7. Agradecimentos

Este trabalho teve apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

Referências

- [Dias and Araújo 2024] Dias, H. D. d. A. and Araújo, J. d. S. (2024). An experience report on the use of active methodologies in building parallel computational thinking. *International Journal of Computer Architecture Education*, 13(1):7–16.
- [Diogo et al. 2023] Diogo, C., Diogo, C., and Santos, V. (2023). Potencialidades do uso de metodologias ativas em disciplinas de computação: Uma revisão sistemática de literatura. In *Anais do XXXIV Simpósio Brasileiro de Informática na Educação*, pages 175–186, Porto Alegre, RS, Brasil. SBC.
- [Fernandes et al. 2023] Fernandes, A., Damasceno, E., and Valentim, N. (2023). Um mapeamento sistemático da literatura sobre ambientes digitais para o treinamento de profissionais da educação. In *Anais do XXXIV Simpósio Brasileiro de Informática na Educação*, pages 496–508, Porto Alegre, RS, Brasil. SBC.
- [Ferreira and Bernardini 2024] Ferreira, J. and Bernardini, F. (2024). Inserindo dados abertos na educação básica: Um mapeamento sistemático da literatura sobre abordagens e recursos educacionais. In *Anais do XXXV Simpósio Brasileiro de Informática na Educação*, pages 486–500, Porto Alegre, RS, Brasil. SBC.
- [Galante 2023] Galante, G. (2023). Experiences with teaching architectures and parallel programming: From construction to evaluation. *International Journal of Computer Architecture Education*, 12(2):40–47.
- [Hyury et al. 2024] Hyury, A., Mercês, S., Coqueiro, T., Ruiz, I., Carvalho, T., and Jailton, J. (2024). Principais aplicações da computação desplugada no ensino fundamental ii: Um mapeamento sistemático da literatura. In *Anais do XXX Workshop de Informática na Escola*, pages 320–330, Porto Alegre, RS, Brasil. SBC.
- [Jr. et al. 2022] Jr., R. A., Farias, H., and Araújo, J. (2022). Um mapeamento sistemático da literatura sobre o ensino e aprendizagem da programação paralela com o uso de placas gráficas. In *Anais do XXX Workshop sobre Educação em Computação*, pages 439–450, Porto Alegre, RS, Brasil. SBC.

- [Perez et al. 2023] Perez, F., Oliveira, W., Lima, A., Junior, A. C., and Araújo, G. (2023). Ensino de programação paralela na educação básica: Uma revisão sistemática da literatura, análise bibliométrica e agenda para estudos futuros. In *Anais do III Simpósio Brasileiro de Educação em Computação*, pages 19–26, Porto Alegre, RS, Brasil. SBC.
- [Saavedra et al. 2024] Saavedra, M. R. M., Aracaty, L. Q., Quaresma, T. M., and Araújo, J. d. S. (2024). Um mapeamento da literatura sobre o uso de jogos sérios na construção do pensamento computacional paralelo. *RENOTE*, 22(1):436–445.