



Sumawma: Uma Plataforma para Auxiliar no Ensino de Códigos Paralelos

Marcos Rodrigo Mendes Saavedra, Instituto de Ciências Exatas e Naturais/UFGA,
marcosaavedraa3@gmail.com, <https://orcid.org/0000-0002-4698-6202>

Thiago de Castro Sant'Anna, Instituto de Ciências Exatas e Naturais/UFGA,
thiago.csantanna04@gmail.com, <https://orcid.org/0009-0006-8249-4906>

José Maria P. Vilhena. Ferreira, Instituto de Ciências Exatas e Naturais/UFGA,
jose.22.vilhena@gmail.com, <https://orcid.org/0009-0001-7089-5060>

Josivaldo de Souza Araújo, Instituto de Ciências Exatas e Naturais/UFGA,
josivaldo@ufpa.br, <https://orcid.org/0000-0001-6890-6923>

Resumo: A construção de códigos paralelos pode ser um desafio no processo de ensino e aprendizagem da Programação Paralela, dependendo do sistema operacional utilizado. Para se obter um ambiente funcional, é necessário antes instalar e configurar as principais bibliotecas paralelas, como OpenMP, OpenMPI e OpenACC. Em seguida, são requeridas ferramentas para editar, compilar e executar os códigos gerados. Para quem não está familiarizado com esses ambientes e configurações, todo esse processo pode desestimular o interesse e até mesmo o aprendizado da Programação Paralela. Este trabalho apresenta a plataforma Sumawma, que tem como objetivo apoiar a elaboração de códigos paralelos e auxiliar no processo de ensino e aprendizagem da Programação Paralela, por meio de um ambiente pré-configurado para o desenvolvimento e a execução dos códigos, sem a necessidade de instalar ou configurar qualquer biblioteca paralela. Além disso, por ser um ambiente *on-line*, pode ser executado em qualquer sistema operacional, via *browser*, com base em uma arquitetura cliente-servidor. O sistema foi testado e avaliado em uma turma de Programação Paralela e, também, por especialistas da área, que aprovaram o recurso didático.

Palavras-chave: Programação Paralela, Ambientes de Programação, Plataforma, Ferramenta, Bibliotecas Paralelas.

Sumawma: A Platform to Assist in the Teaching of Parallel Codes

Abstract: *The development of parallel code can be challenging in the teaching and learning process of Parallel Programming, depending on the operating system used. To obtain a functional environment, it is first necessary to install and configure major parallel libraries, such as OpenMP, OpenMPI, and OpenACC. Subsequently, tools are required to edit, compile, and execute the generated code. For those unfamiliar with these environments and configurations, this entire process can discourage interest and even hinder the learning of Parallel Programming. This paper presents the Sumawma platform, which aims to support the development of parallel code and assist in the teaching and learning process of Parallel Programming through a pre-configured environment for code development and execution, eliminating the need to install or configure any parallel library. Furthermore, being an online environment, it can run on any operating system via a browser, based on a client-server architecture. The system was tested and evaluated in a Parallel Programming class and also by experts in the field, who approved the educational resource.*

Keywords: *Parallel Programming, Programming Environments, Platform, Tool, Parallel Libraries.*

1. Introdução

A Programação Paralela se estabeleceu como uma disciplina essencial para a área da Computação. Não apenas devido ao aumento da necessidade de se processar



grandes quantidades de dados, em aplicações cada vez mais complexas, mas também devido à sua inserção nas matrizes curriculares por recomendação de entidades como o *IEEE (Institute of Electrical and Electronics Engineers)*, a *ACM (Association for Computing Machinery)* e a *SBC (Sociedade Brasileira de Computação)* (SAAVEDRA *et al.*, 2024). Essas entidades ressaltam a relevância, cada vez maior, do paralelismo na Educação em Computação (DRAFT, 2013). Contudo, seu processo de aprendizagem ainda enfrenta obstáculos consideráveis, que vão desde a abstração de conceitos, como concorrência e sincronização, passando pela necessidade de instalar e configurar o ambiente, utilizando as bibliotecas paralelas OpenMP, MPI e OpenACC, e a falta de uma infraestrutura apropriada nas instituições de ensino. Esses obstáculos, além de proporcionarem uma elevada taxa de evasão nos cursos de graduação, podem proporcionar, também, problemas na compreensão do conteúdo.

Para contornar parte desses obstáculos, metodologias e ferramentas são utilizadas com o objetivo de apoiar o processo de ensino e aprendizagem da Programação Paralela, tornando-o mais simples, compreensível e acessível. Metodologias como a Gamificação, Jogos Educacionais, elaboração de projetos facilitam a compreensão de conceitos abstratos e envolvem os alunos em um ambiente mais interativo e prático (SOUZA; SAAVEDRA e ARAÚJO, 2023). Por outro lado, a utilização de ambientes que já venham configurados para a elaboração, execução e compilação de códigos paralelos acaba auxiliando, minimizando erros e facilitando o trabalho de preparação, que muitas vezes, dependendo do sistema operacional utilizado, pode resultar em mais ou menos esforço de configuração.

Com o objetivo de reduzir as dificuldades e erros de configuração, bem como a dependência dos sistemas operacionais, este trabalho apresenta um ambiente *on-line* para edição, compilação e execução de códigos paralelos. Neste ambiente, as principais ferramentas paralelas (OpenMP, OpenMPI e OpenACC) já estão disponíveis (instaladas e configuradas) para serem executadas em programas escritos nas linguagens C/C++. Diante desse contexto, este estudo busca responder à seguinte pergunta de pesquisa: Como uma plataforma web pré-configurada pode reduzir as barreiras iniciais de configuração de ambiente e melhorar a experiência de ensino-aprendizagem em programação paralela em cursos de graduação em Computação?

O ambiente foi desenvolvido e, posteriormente, avaliado em uma turma de Tópicos Especiais em Programação Paralela em um curso de graduação em Computação e, também, por especialistas na área de Programação. Na avaliação, foram apontados os principais pontos positivos e negativos. Os resultados foram motivadores, observando-se uma aceitação por parte dos avaliadores (alunos e professores) do ambiente proposto, pois, como o mesmo já se apresenta configurado, não há a necessidade de se instalar ou configurar qualquer outro *software*.

2. Trabalhos Relacionados

Foi realizado um mapeamento sistemático da literatura como levantamento bibliográfico sobre o uso de ferramentas web que auxiliam no processo de ensino de Programação Paralela (SAAVEDRA *et al.*, 2025). Este trabalho identificou treze ferramentas desenvolvidas para o ensino da Programação Paralela que utilizam diversas linguagens de programação e *frameworks* de paralelização, como o OpenMP, MPI e CUDA, além de diferentes conceitos e estratégias pedagógicas associadas ao tema. O mapeamento analisou aspectos como disponibilidade da ferramenta, metodologias de ensino adotadas, mecanismos de avaliação e *feedback*.

Embora os resultados evidenciem avanços no apoio ao ensino de Programação



Paralela, o estudo também aponta lacunas relacionadas à integração com o conteúdo abordado. Este trabalho se diferencia dos demais encontrados, pois foi aplicada a utilização dos conceitos de Programação Paralela em sala de aula, com o uso de uma metodologia baseada em projetos, a partir do auxílio da plataforma Sumawma para as atividades práticas, utilizando os *frameworks* OpenMP, MPI e OpenACC. Dessa forma, os alunos puderam aplicar os conceitos de Paralelização programando funcionalidades com Processos e *Threads*. A plataforma foi avaliada tanto por alunos, quanto por especialistas, onde puderam apontar as vantagens e eficácia da plataforma em sala de aula, bem como, sugerir melhorias, as quais foram inseridas na versão final do Sumawma.

3. O Ensino da Programação Paralela

Para a consolidação da Programação Paralela como disciplina, foi fundamental a sua inserção nas matrizes curriculares dos cursos de graduação em Computação. Esse processo foi iniciado em 2013 pela ACM e IEEE por meio do documento “Diretrizes Curriculares para Programas de Graduação em Ciência da Computação”, que propôs regulamentações no ensino da Programação Paralela para os currículos dos cursos de Ciência da Computação (DRAFT, 2013). Essas diretrizes definem, a partir dos conceitos aplicados em sala, as seguintes habilidades a serem formadas nos estudantes:

1. Fundamentos de Paralelismo: distinguir paralelismo de concorrência e identificar as condições de corrida de dados e de nível superior;
2. Decomposição Paralela: paralelizar algoritmos usando o conceito de decomposição de tarefas/dados e escrever programas com atores;
3. Comunicação e Coordenação: fazer uso de exclusão mútua, evitar *deadlocks* e implementar filas sincronizadas;
4. Algoritmos Paralelos: calcular trabalhos, implementar algoritmos paralelos e medir o desempenho desses algoritmos.
5. Arquitetura Paralela: diferenciar memória compartilhada/distribuída e descrever as arquiteturas paralelas;
6. Desempenho Paralelo: otimizar desempenho, detectar compartilhamento falso e avaliar o consumo de energia.

Já a Sociedade Brasileira de Computação (SBC) engloba a Programação Paralela dentro do eixo de formação referente ao Desenvolvimento de Sistemas Computacionais, buscando promover soluções computacionais eficientes para diversos tipos de problemas, envolvendo *hardware*, *software* e processos.

3.1. Dificuldades no Ensino da Programação Paralela

A partir desta descrição para a formação de habilidades aos egressos do curso de Computação, tem-se pensado em técnicas que solucionem os problemas enfrentados para o ensino da Programação Paralela, já que um dos principais fatores é o grau de dificuldade para solucionar problemas a partir das regras de paralelização (DALEIDEN *et al.*, 2020). Para isso, é importante a aplicação de metodologias que auxiliem na compreensão desses conceitos, como, por exemplo, a adoção de metodologias ativas de ensino.

Outro problema enfrentado no ensino da Programação Paralela é a falta de laboratórios próprios para que os discentes possam executar os exemplos aplicados em sala de aula (NETO; NETO e MORENO, 2022). Algumas metodologias têm sido usadas para enfrentar estas dificuldades, como a adoção de atividades “*desplugadas*”, como explica (DIAS e ARAÚJO, 2024). A própria SBC aponta como um desafio o Desenvolvimento de Recursos Pedagógicos para o ensino da Computação no Ensino Básico (SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2025). Contudo, essa



necessidade também se reflete nos cursos de graduação e é necessário, dessa forma, estabelecer metas e empregar formas para conter o alto índice de evasão. Para isso, é necessário adotar ferramentas que possam contribuir, não apenas para que o ensino da Programação Paralela, mas também para o ensino da Programação de uma forma geral, para que o aprendizado possa ser estimulado de forma eficiente em sala de aula.

4. Metodologia da Pesquisa

Este trabalho adotou uma abordagem avaliativa e exploratória, com foco no desenvolvimento e na análise de usabilidade e da aceitação da plataforma Sumawma no contexto educacional, tendo os conceitos de Programação Paralela como objeto de estudo. O método foi estruturado em três etapas:

1. Desenvolvimento da Plataforma: esta etapa utilizou diversas ferramentas para o desenvolvimento de uma arquitetura cliente-servidor que foi configurada em um contêiner Docker com os principais recursos para executar as bibliotecas paralelas, como banco de dados PostgreSQL, servidor Apache, NodeJS e ReactJS, assim como o pacotes de configuração para processamento em paralelização MPI e GPU. A descrição completa do desenvolvimento pode ser visualizada no Item 5.
2. Avaliação com Usuários (Estudantes): A avaliação com estudantes foi realizada na disciplina de Tópicos Especiais de Programação Paralela e ao final da disciplina, responderam ao formulário de avaliação. As avaliações realizadas com os estudantes estão descritas no Item 6, o perfil dos participantes e os resultados estão nos Itens 7.1 e 7.3.
3. Avaliação com Especialistas: Cinco professores com experiência em programação ou ensino de programação paralela avaliaram a plataforma seguindo um roteiro estruturado. A metodologia de avaliação com os professores especialistas pode ser acompanhada no Item 6, o perfil dos especialistas está descrito no Item 7.2 e os resultados no Item 7.4.

A análise dos dados combinou métodos quantitativos (cálculo do *score* SUS para medir a usabilidade pelos estudantes) e qualitativos (análise das respostas abertas e sugestões, tanto dos estudantes que responderam as questões, quanto dos professores especialistas). A análise desses dados permitiu uma avaliação técnica dos recursos propostos pela plataforma Sumawma, bem como uma avaliação do seu potencial impacto para a sociedade.

5. Descrição da Plataforma

A Plataforma Sumawma é uma ferramenta para auxiliar a aprendizagem da Programação Paralela, permitindo que os usuários possam escrever os seus códigos através das linguagens de programação C, utilizando as principais bibliotecas de paralelização *OpenMP*^{*}, *OpenMPI*[†] e *OpenACC*[‡].

Esta plataforma utiliza uma arquitetura cliente-servidor, onde os usuários acessam a interface Web para criação (edição) dos códigos, podendo em seguida realizar a compilação e a execução dos códigos criados. Ou seja, executam todas as etapas sem ter a necessidade de mudar de ambiente ou digitar qualquer comando, tudo realizado através da própria interface da plataforma. Após a execução, o resultado é exibido ao usuário, também, via interface. A proposta da plataforma Sumawma é simplificar o processo de edição, compilação, execução e apresentação dos resultados. Dessa forma, os alunos não necessitam memorizar comandos, mudar de tela ou abrir arquivos.

^{*}<https://www.openmp.org/>

[†]<https://www.open-mpi.org/>

[‡]<https://www.openacc.org/>



Para este trabalho, o servidor da plataforma Sumawma foi executado em um computador que possuía uma arquitetura de CPU com 8 núcleos de processamento, memória RAM de 8 *Gigabytes* e uma Placa de Vídeo da Nvidia, modelo *Nvidia GeForce GTX 1660*, com 1.280 núcleos de processamento e 6 *Gigabytes* de memória. O sistema operacional usado, no servidor, foi o Linux com a distribuição Ubuntu 22.02.

O projeto da plataforma Sumawma foi desenvolvido utilizando duas ferramentas: o *ReactJS* e o *NodeJS*. A primeira é uma biblioteca do *JavaScript* para criação de interface de usuário. A partir dessa ferramenta, foram desenvolvidas as telas de interação do usuário, seguindo critérios de acessibilidade[§] do WCAG (*Web Content Accessibility Guidelines*)[¶], permitindo que um maior número de pessoas possível consiga utilizar a plataforma e todas as ferramentas que ela disponibiliza.

Já o ambiente, que é executado no servidor, foi desenvolvido em *NodeJS*, que é um *framework*, também do *Javascript*, e é utilizado para o desenvolvimento de sistemas *Backend*. O servidor realiza uma comunicação com o *PostgreSQL* que é um sistema de banco de dados relacional. No banco de dados são armazenados os dados de acesso dos alunos (*login* e senha), e os dados dos projetos criados por cada um dos usuários (cada usuário só tem acesso aos seus próprios projetos).

O projeto utiliza ainda um *contêiner Docker* que possui a configuração do *ReactJS*, do projeto Node (incluindo as bibliotecas usadas), o sistema de banco de dados *postgresql*, os pacotes e os *drivers* necessários para a configuração. Além disso, possui também o funcionamento dos *frameworks* de paralelização e o *Apache Server*, para tornar a interface disponível na rede, para cada um dos usuários da plataforma.

Quando o usuário cria um projeto por meio da interface, um diretório é criado no servidor e os arquivos do projeto são armazenados dentro do diretório criado, permitindo que o usuário acesse o ambiente já configurado. Os projetos desenvolvidos ficam disponíveis na plataforma Sumawma, onde podem ser acessados a qualquer momento. O usuário tem a opção de criar vários arquivos dentro de cada projeto criado. Também pode editar e excluir os arquivos contidos nos projetos.

Além disso, o usuário pode utilizar a quantidade de núcleos de processamento disponíveis no ambiente servidor. Para o OpenMP e o OpenACC, essa definição de quantidade de núcleos pode ser definida no código, contudo, o OpenMPI utiliza *flags* na linha de comando, que define o número de processos que vão ser utilizados para executar o código. Como solução para este problema, foi criado um campo na interface do editor do OpenMPI que permite ao usuário definir esse número de processos (Figura 1).

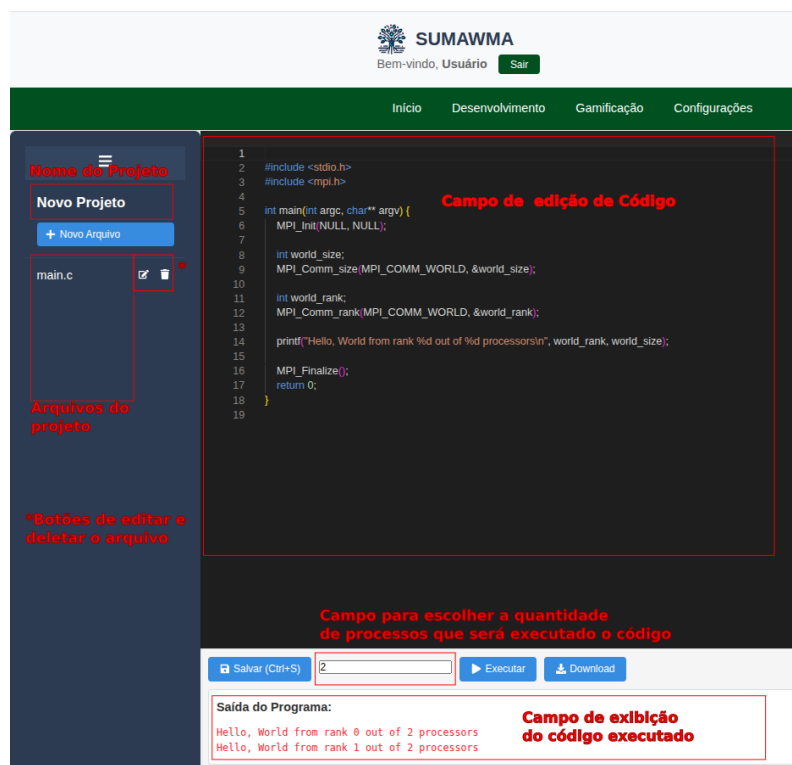
6. Metodologia Utilizada na Avaliação da Plataforma

A plataforma Sumawma foi utilizada em uma disciplina de Tópicos Especiais em Programação Paralela de um curso de graduação em Ciência da Computação. A metodologia consistiu em medir o impacto da plataforma e sua contribuição para o ensino da Programação Paralela. Para isso, foi realizada uma pesquisa de campo com estudantes e professores especialistas da instituição, por meio de questionários que identificaram pontos fortes e fracos, lacunas na adoção da plataforma Sumawma, bem como sua usabilidade.

Para mensurar a usabilidade, adotou-se a metodologia *System Usability Scale* (SUS), amplamente utilizada para avaliar a usabilidade de produtos de software e hardware (HYZY *et al.*, 2022). A avaliação pelo SUS foi realizada por meio de um questionário com dez perguntas, aplicadas em uma ordem específica. As respostas das

[§] Critérios WCAG: [1.1.1], [1.3.1], [1.4.1], [1.4.3], [1.4.4], [2.1.1], [2.3.1], [2.4.4], [2.4.6], [3.1.1], [3.3.2]

[¶] <https://guia-wcag.com/>

Figura 1. Tela do Editor de Código *OpenMPI*

perguntas podem ser visualizadas na Figura 2. Cada pergunta tem cinco alternativas, baseadas na escala Likert, que varia de 1 (discordo totalmente) a 5 (concordo totalmente) (GERALDES; MARTINS e AFONSECA, 2019). O cálculo da usabilidade do software segue as seguintes etapas:

1. Para cada participante $X = (\text{Soma das perguntas ímpares}) - 5$;
2. Para cada participante $Y = 25 - (\text{Soma das perguntas pares})$;
3. $Z = (X + Y) * 2,5$ para cada participante;
4. É realizada a média do valor de Z de todos os participante da pesquisa.

A avaliação do método SUS pode classificar a usabilidade a partir da escala de 0 a 100 (COSTA; SANTOS e SANTIAGO, 2021), a faixa de classificação pode ser consultada na Tabela 1.

Tabela 1. Tabela de classificação do SUS.

<49,9	50,0 - 59,9	60,0 - 69,9	70,0 - 79,9	80,0 - 89,9	90,0 - 100
Muito Ruim	Ruim	Ok	Bom	Excelente	Melhor Imaginável

A avaliação com os alunos foi realizada no final do semestre, após terem utilizado a plataforma durante a disciplina. Para isso, foram submetidos a um questionário¹, contendo dez perguntas sobre a usabilidade, seis perguntas sobre a experiência em sala e quatro perguntas abertas, onde podiam dar suas opiniões livremente. Para as perguntas de usabilidade e experiência em sala, como já citado, foi utilizada a Escala *Likert*.

A avaliação com os professores especialistas foi realizada com o objetivo de obter um retorno da utilidade da plataforma no processo de ensino e aprendizagem da Programação Paralela. Para isso, foi desenvolvido um roteiro**que deveria ser seguido

¹Formulário para os alunos: <https://shre.ink/oymt>.

^{**}Roteiro de testes com especialistas: <https://shre.ink/o0UG>.

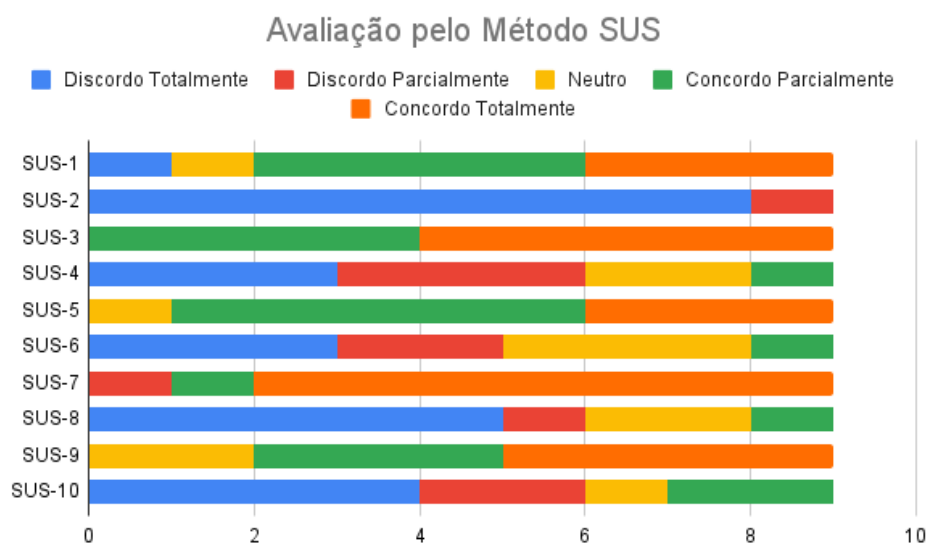


Figura 2. Avaliação sobre Usabilidade realizada pelos Alunos

para a avaliação. Em seguida, os professores responderam a um questionário^{††} avaliando a plataforma. Esse formulário foi organizado em quatro classes de perguntas: Informações sobre o pesquisador, o uso da plataforma no aprendizado da Programação Paralela, a usabilidade da plataforma e possíveis contribuições do especialista à pesquisa.

6.1. Descrição da Disciplina de Tópicos em Programação Paralela

O objetivo principal da disciplina foi fazer com que o aluno conseguisse identificar a necessidade de reduzir o tempo computacional de problemas que exigem um grande poder de processamento e uma grande quantidade de memória. Isso é possível através da decomposição do problema em subproblemas, de forma que se possa estabelecer uma comunicação entre eles. Como objetivos específicos:

1. Proporcionar ao aluno uma compreensão sobre os princípios nos quais está baseada a Programação Paralela, bem como as ferramentas e as bibliotecas que são necessárias para se conseguir prover esse tipo de programação;
2. Compreender as linguagens, algoritmos e projetos que servem como base para o desenvolvimento de códigos para os sistemas de alto desempenho;
3. Analisar características como desempenho e estrutura de códigos paralelos;

A disciplina abordou os princípios fundamentais da Programação Paralela, com enfoque em tópicos como Arquiteturas Paralelas, Modelos Hierárquicos, Padrões de Programação Paralela e o uso de Bibliotecas amplamente utilizadas na área, como OpenMP, MPI e OpenACC. As aulas da disciplina foram ministradas em duas etapas: teórica e prática, a fim de proporcionar uma maior absorção do conteúdo e capacitar os estudantes no desenvolvimento de algoritmos e códigos paralelos.

7. Resultados e Discussões

Durante a disciplina foi utilizada e avaliada a plataforma Sumawma, para a criação, compilação e execução de códigos a partir dos *frameworks* apresentados para os diversos tipos de arquiteturas paralelas. Foi avaliado o perfil dos participantes (curso e períodos), a eficiência da plataforma e possíveis sugestões.

^{††}Formulário para os especialistas: <https://shre.ink/oEiP>.



7.1. Perfil dos alunos Participantes

A disciplina foi realizada por 16 estudantes de três cursos: Ciência da Computação, Sistemas de Informação e Engenharia da Computação. Contudo, prosseguiram até o final da disciplina apenas 12 participantes: destes, 9 responderam ao formulário de avaliação (três alunos de Ciência da Computação do sexto período e quatro do oitavo; um aluno de Engenharia da Computação e um aluno de Sistemas de Informação do nono período).

Como a disciplina foi ofertada como parte da matriz curricular do curso de Ciência da Computação, reflete o maior número de integrantes deste curso. Os discentes, dos outros cursos, relataram que se matricularam na turma pela necessidade de aprender os conceitos da Programação Paralela para utilizarem nos projetos em que estavam atuando, e também pela disciplina não ser ofertada em suas respectivas matrizes curriculares. Avalia-se que todos os participantes da pesquisa já possuíam uma base sólida de programação. Isso pode ser concluído pelo fato de que os mesmos já terem integralizado mais de 75% dos seus cursos (considerando que o curso de Ciência da Computação tem 8 períodos, Sistemas de Informação possui 9 períodos e Engenharia da Computação 10 períodos).

7.2. Perfil dos Participantes Especialistas

A pesquisa contou com a participação de cinco professores especialistas. O tempo de experiência docente dos participantes variou entre 7 e 27 anos. Todos os cinco professores têm experiência na docência da disciplina de Programação há pelo menos 5 anos, e quatro dos avaliadores desenvolveram ou desenvolvem trabalhos com a Programação Paralela (ensino, pesquisa e extensão) há pelo menos 3 anos.

Com relação ao grau de conhecimento em Programação Paralela, em uma escala de 1 (menor) a 5 (maior): Um professor se avaliou com “1”, dizendo não ter familiaridade no assunto, três professores responderam com “4”, relatando que têm conhecimento sobre o tema, e 1 professor respondeu com “5”, relatando que possui total domínio da Programação Paralela. O professor que se avaliou com “1”, apesar de não possuir familiaridade com a Programação Paralela, é um especialista na área da Interação Humano-Computador, sendo, portanto, fundamental na avaliação do ambiente como um todo.

Os professores, também, foram perguntados “Qual foi o ambiente/ferramenta utilizado para desenvolver ou ensinar Programação (sequencial) e/ou Programação Paralela?”. Diversas ferramentas foram citadas, como: a linguagem R, OpenMPI, Pthreads, Java, e ambientes como *Pycharm*, *Google Collab*, VS Code, Dev C++ e Eclipse. Essa pergunta foi utilizada para analisar a familiaridade dos professores com plataformas similares ao Sumawma.

7.3. Avaliação da Plataforma Realizada pelos Estudantes

Foram calculadas as pontuações a partir das respostas de cada participante e coletada a média desses resultados conforme a classificação do método SUS. O resultado obtido foi 79.72, sendo classificado como “BOM”(Tabela 1). Este resultado aponta que a interface foi considerada de fácil uso e que os estudantes conseguiram, de maneira simples, executar todas as tarefas propostas durante as aulas.

Pelo gráfico da Figura 3, pode-se perceber que a plataforma Web teve uma avaliação positiva como uma ferramenta de apoio no processo de ensino e aprendizagem da Programação Paralela, mas especificamente, nos processos de criação, compilação e execução dos códigos paralelos. Pois, evita com que os alunos necessitem instalar



e/ou configurar qualquer software e, principalmente, não dependam de qualquer sistema operacional que esteja sendo utilizado nos laboratórios de ensino. Percebe-se, também, que a pergunta 5 foi a que mostrou maior divergência entre os participantes. Isso porque essa pergunta diz respeito à identificação dos erros no programa. Porém, com o *feedback* dos alunos, esse problema foi corrigido.

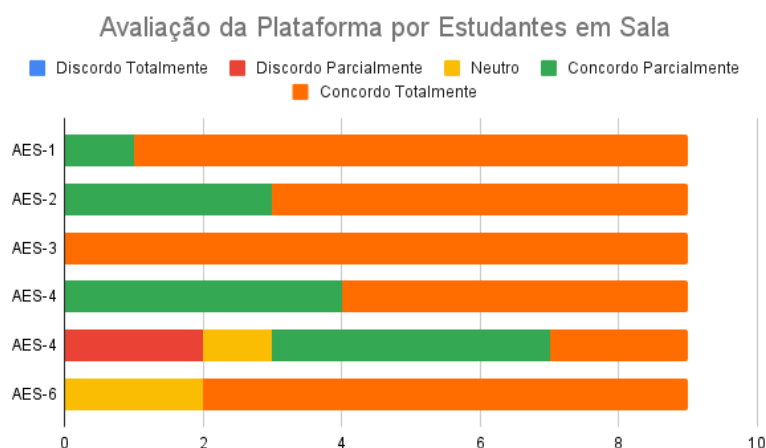


Figura 3. Avaliação da eficiência do uso da plataforma em sala pelos alunos

As quatro questões de livre opinião possibilitaram uma melhor compreensão da percepção dos alunos quanto ao uso da plataforma, pois puderam apontar pontos positivos e negativos, além de sugerir melhorias^{‡‡}

Para a **Questão 1**, sobre os desafios relatados, foram solucionados no decorrer da disciplina. A questão de “salvar” os códigos, relatados nas (**respostas de 2 a 5**), foi observada que criar um atalho no teclado para o salvamento dos códigos facilitava o uso. Para isso, foi implementado o “CTRL+S” para o botão “Salvar”.

Já o problema citado no tempo de expiração da sessão, (**resposta 6**), era referente ao *token* de autenticação, que tinha como prazo de duração inicial apenas 1 hora. No entanto, foi alterado para 3 horas, evitando que o mesmo expirasse antes do final da aula. Já na (**resposta 8**), ocorria devido à implementação de uma funcionalidade de segurança, onde são “trancadas” as rotas após o *login* na plataforma, contudo, foi observada a necessidade de liberar algumas dessas rotas, evitando com que o usuário tenha que se autenticar novamente.

Com relação à forma de exibição dos erros de compilação, (**resposta 9**) onde os usuários precisavam entender melhor para realizar a manutenção dos algoritmos, foi implementado um retorno dos erros de forma mais explícita. Facilitando o reconhecimento e tratamento por parte dos usuários.

Para a **Questão 2** referente a acessibilidade, todos os participantes afirmaram que não tiveram necessidade de outros recursos de acessibilidade além dos já implementados, porém, a plataforma não foi testada com pessoas com deficiência, o que pode haver a necessidade de implementação de novos recursos de acessibilidade.

Na **Questão 3**, foram destacados os pontos positivos da plataforma, como “permitir praticar o conteúdo teórico em sala”, “o ambiente ser de fácil uso” e “permitir a fluidez do conteúdo”, possibilitando que o cronograma de aula fosse cumprido na sua totalidade. Esses pontos relatados reforçam a hipótese levantada pela pesquisa, através do método de observação, devido à falta de laboratórios específicos de ensino da

^{‡‡}Respostas das questões feitas aos estudantes: <https://shre.ink/oRMh>.



Programação Paralela e ao tempo gasto configurando o ambiente de desenvolvimento. Essas ações não permitiam (ou dificultavam) seguir o cronograma da disciplina e abordar as principais arquiteturas paralelas, principalmente o aprendizado das arquiteturas *multicore* e *manycore*.

A aplicação da plataforma em sala de aula contou com uma primeira versão que foi melhorada com alguns ajustes necessários para os alunos. Mas, mesmo assim, a plataforma em sua primeira versão se demonstrou eficiente. Na avaliação realizada, os estudantes consideraram o nível de interação com a plataforma fácil e conseguiram desenvolver seus códigos sem grandes dificuldades, a partir dos conteúdos ministrados em sala. A avaliação completa realizada pelos estudantes pode ser visualizada no seguinte documento: <https://shre.ink/oy1j>

7.4. Avaliação Realizada pelos Professores Especialistas

A Avaliação Com Especialistas (ACE) da plataforma como recurso para o ensino da Programação Paralela, reflete que a plataforma, no geral, é considerada uma ferramenta capaz de auxiliar no processo de ensino. Isso, devido a 80% das respostas (sete das perguntas) serem consideradas positivas (concordaram parcialmente e totalmente) e três das perguntas foram respondidas por todos os professores que concordaram totalmente, conforme Figura 4.

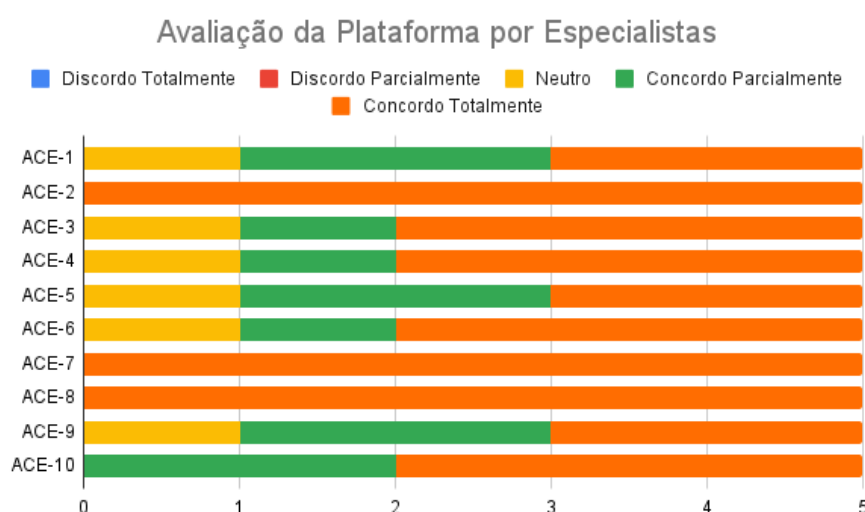


Figura 4. Avaliação da plataforma como ferramenta de auxílio ao ensino da Programação Paralela realizada pelos Professores Especialistas

A Avaliação da Usabilidade por Especialistas (AUP), de acordo com o gráfico da Figura 5, retrata que a plataforma possui uma interface organizada e com suas funcionalidades de fácil aprendizado. Um dos avaliadores relatou que é preciso melhorar alguns elementos de interação nas telas de tutoriais e exemplos. Alguns especialistas se consideraram inaptos a avaliar os recursos de acessibilidade (conforme a AUP-7 da Figura 5). Um avaliador recomendou desenvolver botões de desfazer e refazer no sistema (atualmente, a plataforma utiliza as funcionalidades do navegador). Os avaliadores destacaram como pontos fortes que a plataforma pode ser usada, no contexto acadêmico, como uma ferramenta de ensino da Programação Paralela e Distribuída, com compiladores de fácil uso e mensagens de erro explicativas. Que a plataforma possui uma IDE com modelos distintos e, principalmente, que não possui a necessidade de configurar o ambiente de desenvolvimento.

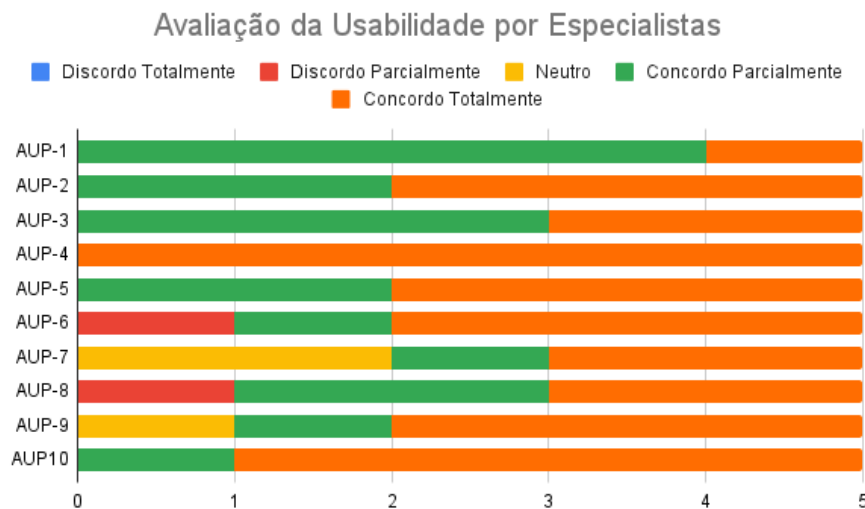


Figura 5. Avaliação da Usabilidade realizada pelos Professores Especialistas

Como pontos fracos, foram apontados: Os exemplos e tutoriais devem ter uma melhor visibilidade (sendo adicionados no menu superior); ao criar um arquivo novo em um projeto, este arquivo já deve ter a extensão automática do arquivo; e a criação de um tutorial dinâmico, ensinando a usar a plataforma, principalmente no primeiro acesso.

A experiência da Plataforma Sumawma com as demais IDEs já utilizadas pelos avaliadores foi considerada satisfatória, pois as interfaces foram consideradas similares, o que facilitou o uso. Porém, foi destacado como vantagem que a Sumawma não precisa configurar um ambiente de desenvolvimento (como as IDEs tradicionais), o que facilita o aprendizado. Os avaliadores, por fim, recomendaram o uso da plataforma para o ensino da Programação Paralela, assim como alguns desejam adotá-la como recurso didático. A avaliação completa realizada pelos professores especialistas pode ser visualizada no seguinte documento: <https://shre.ink/oy1t>.

8. Conclusão

O estudo da Programação Paralela é considerado um desafio nas instituições de ensino, devido a problemas como a complexidade de conceitos fundamentais e a necessidade de uma infraestrutura e ambientes adequados. Este artigo apresentou o uso da plataforma Sumawma onde se pode escrever, compilar e executar códigos paralelos, desenvolvidos com *frameworks* do OpenMP, OpenMPI e OpenACC sem a necessidade de configurar o ambiente.

Os resultados apontaram que uma abordagem prática, aliada a uma plataforma Web para desenvolvimento e execução de códigos paralelos, não apenas facilita o entendimento dos conceitos, mas também incentiva a participação ativa dos alunos. Ao alinhar-se com as diretrizes da ACM, IEEE e SBC, este trabalho reforça a importância de integrar a Programação Paralela nos currículos de Computação, preparando os futuros profissionais para os desafios da computação heterogênea e de alto desempenho.

Como trabalhos futuros, tem-se a expansão da plataforma com melhorias na depuração dos resultados, mensagens de erro mais explícitas e a inclusão de metodologias, como jogos educacionais, *quizzes* e atividades em grupo. Além disso, também serão realizados testes com pessoas com deficiência para avaliar os recursos de acessibilidade



implementados, que não puderam ser realizados devido a falta de participantes que precisassem de recursos de acessibilidade.

9. Agradecimentos

Este trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES).

Referências

- COSTA, R.; SANTOS, A.; SANTIAGO, C. Análise de usabilidade do sistema q-acadêmico utilizando o método system usability scale (sus): Um estudo de caso. In: **Anais do XIV Encontro Unificado de Computação do Piauí e XI Simpósio de Sistemas de Informação**. Porto Alegre, RS, Brasil: SBC, 2021. p. 231–238. ISSN 0000-0000. Disponível em: [⟨https://sol.sbc.org.br/index.php/enucompi/article/view/17776⟩](https://sol.sbc.org.br/index.php/enucompi/article/view/17776).
- DALEIDEN, P.; STEFIK, A.; UESBECK, P. M.; PEDERSEN, J. Analysis of a randomized controlled trial of student performance in parallel programming using a new measurement technique. **ACM Transactions on Computing Education (TOCE)**, ACM New York, NY, USA, v. 20, n. 3, p. 1–28, 2020.
- DIAS, H. D. d. A.; ARAÚJO, J. d. S. An experience report on the use of active methodologies in building parallel computational thinking. **International Journal of Computer Architecture Education**, v. 13, n. 1, p. 7–16, Dec. 2024. Disponível em: [⟨https://journals-sol.sbc.org.br/index.php/ijcae/article/view/5339⟩](https://journals-sol.sbc.org.br/index.php/ijcae/article/view/5339).
- DRAFT, S. Computer science curricula 2013. **ACM and IEEE Computer Society, Incorporated: New York, NY, USA**, 2013.
- GERALDES, W.; MARTINS, E.; AFONSECA, U. Avaliação da usabilidade do scratch utilizando o método system usability scale (sus). In: **Anais da X Escola Regional de Informática de Mato Grosso**. Porto Alegre, RS, Brasil: SBC, 2019. p. 25–30. ISSN 2447-5386. Disponível em: [⟨https://sol.sbc.org.br/index.php/eri-mt/article/view/8589⟩](https://sol.sbc.org.br/index.php/eri-mt/article/view/8589).
- HYZY, M. *et al.* System usability scale benchmarking for digital health apps (preprint). **Jmir mhealth and uhealth**, 2022.
- NETO, J. A. C.; NETO, A. J. A.; MORENO, E. A systematic review on teaching parallel programming. **Proceedings of the 11th Euro American Conference on Telematics and Information Systems**, 2022.
- SAAVEDRA, M.; QUARESMA, T.; OLIVEIRA, M.; ARAÚJO, J. Um mapeamento da literatura sobre o uso de ferramentas para auxiliar no processo de ensino e aprendizagem da programação paralela. In: **Anais do XXXVI Simpósio Brasileiro de Informática na Educação**. Porto Alegre, RS, Brasil: SBC, 2025. p. 746–758. ISSN 0000-0000. Disponível em: [⟨https://sol.sbc.org.br/index.php/sbie/article/view/38464⟩](https://sol.sbc.org.br/index.php/sbie/article/view/38464).
- SAAVEDRA, M. R. M.; ARACATY, L. Q.; QUARESMA, T. M.; Araújo, J. d. S. Um mapeamento da literatura sobre o uso de jogos sérios na construção do pensamento computacional paralelo. **RENOTE**, v. 22, n. 1, p. 436–445, jul. 2024. Disponível em: [⟨https://seer.ufrgs.br/index.php/renote/article/view/141569⟩](https://seer.ufrgs.br/index.php/renote/article/view/141569).
- SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. **Grande Desafios da Computação 2025-2035**. Porto Alegre: SBC, 2025. 32 p. Acesso em: 16 Abril 2025. Disponível em: [⟨https://books-sol.sbc.org.br/index.php/sbc/catalog/book/17⟩](https://books-sol.sbc.org.br/index.php/sbc/catalog/book/17).
- SOUZA, F. R. A. d.; SAAVEDRA, M. R. M.; ARAÚJO, J. d. S. Um mapeamento da literatura sobre o uso de metodologias ativas na construção do pensamento computacional paralelo. **RENOTE**, v. 21, n. 1, p. 57–66, jul. 2023. Disponível em: [⟨https://seer.ufrgs.br/index.php/renote/article/view/134323⟩](https://seer.ufrgs.br/index.php/renote/article/view/134323).