

MFUP: Complementando o aprendizado em Fundamentos de Programação por Meio de Competição

**Samuel C. do Nascimento¹, Beatriz N. de Oliveira¹, Franciel S. P. de Vasconcelos¹,
Wladimir A. Tavares², Rainara M. Carvalho¹, Enyo J. T. Gonçalves¹**

¹ Universidade Federal do Ceará (UFC) - Campus Quixadá
Av. José de Freitas Queiroz, 5003 – Cedro, 63902-580 – Quixadá – CE

²Universidade Federal do Ceará (UFC) - Campus do Pici
Av. Mister Hull, s/n - Pici – 60455-760 – Fortaleza – CE

{samuelc, francielsilveira, beatriznascimento}@ufc.alu.br

{wladimirtavares, rainara, enyo}@ufc.br

Abstract. *Introductory programming courses have high failure rates due to students' lack of familiarity with fundamental concepts. To mitigate this issue, PET-SI developed the Programming Fundamentals Marathon (MFUP), a project based on competitive learning inspired by the Brazilian Computer Society (SBC) competitions. The initiative involved first-semester Information Systems students and used the Beecrowd platform to conduct collaborative challenges. The results indicated a positive impact on participants' motivation and learning, with 76% satisfaction and 42% reporting a high level of contribution to learning the C language.*

Resumo. *As disciplinas introdutórias de programação apresentam elevados índices de reprovação devido à falta de familiaridade dos estudantes com conceitos fundamentais. Para mitigar esse problema, o PET-SI desenvolveu a Maratona de Fundamentos de Programação (MFUP), um projeto baseado em aprendizado competitivo, inspirado nas competições da Sociedade Brasileira de Computação (SBC). A iniciativa envolveu alunos do primeiro semestre de Sistemas de Informação e utilizou a plataforma Beecrowd para a realização de desafios colaborativos. Os resultados indicaram um impacto positivo na motivação e aprendizado dos participantes, com 76% de satisfação e 42% relatando um alto nível de contribuição na aprendizagem da linguagem C.*

1. Introdução

As disciplinas de programação são fundamentais nos cursos superiores da área de computação. No entanto, o ensino de programação representa um grande desafio, especialmente para alunos ingressantes que não possuem experiência prévia com conceitos de lógica e algoritmos. Segundo [Piteira and Costa 2013], ”a programação pode ser um processo desafiador para programadores iniciantes. Eles precisam desenvolver habilidades de memorização, compreensão, resolução de problemas, abstração e pensamento lógico, além de conhecimentos declarativos e procedimentais, entre outros.”

As dificuldades no ensino de programação resultam em altos índices de reaprovação e abandono em disciplinas introdutórias, o que pode impactar negativamente a motivação e o desempenho acadêmico dos estudantes. Bosse e Gerosa [Bosse and Gerosa 2017] realizaram um levantamento sobre os cursos introdutórios de programação na Universidade de São Paulo (USP), indicando que a taxa de insucesso supera 25%. Da mesma forma, Raposo e Guerra [Raposo and Dantas 2016] relatam que, na Universidade Federal da Paraíba (UFPB), o índice de reaprovação (por falta e por nota) atinge 64% nos três cursos da área de Computação. As reprovações em disciplinas de programação estão entre os fatores que contribuem para a evasão nesses cursos. Assim, a adoção de estratégias para mitigar esse problema é bem-vinda.

Diversos fatores podem estar relacionados às reprovações e aos abandonos em Fundamentos de Programação. Dentre eles, podemos mencionar o tempo de estudo dos alunos, o engajamento e a motivação nas abordagens de ensino utilizadas [de Sousa and Leite 2020, Giraffa and da Costa Mora 2013]. [Raposo and Dantas 2016] destacam que a aprendizagem em programação exige estudo contínuo e dedicação à resolução de um grande volume de problemas, a fim de internalizar a lógica e a abstração necessárias. Além disso, observa-se que muitos alunos ingressantes na universidade ainda não possuem hábitos de estudo bem estabelecidos. Dado esse contexto, abordagens complementares à carga horária da disciplina têm sido utilizadas para auxiliar no aprendizado, na tentativa de reduzir as reprovações e os abandonos. Como iniciativas com o intuito de colaborar para a melhoria do aprendizado em programação, podemos citar o uso de Ambientes Virtuais de Aprendizagem (AVA), como o Moodle, com ferramentas de autocorreção automática e integração com outros juízes online [Maciel et al. 2013, Venero and Mena-Chalco 2019], monitoria em sala e extra-sala de maneira permanente, aprendizagem baseada em projetos e o uso de gamificação por meio de competições de programação.

A abordagem pedagógica construtivista enfatiza que o aprendiz constrói ativamente seu próprio conhecimento, em vez de simplesmente assimilar cópias exatas do entendimento do professor. Dentro desse paradigma, a aprendizagem baseada em problemas se destaca ao promover maior engajamento por meio da resolução de desafios contextualizados, incentivando a reflexão e a aplicação prática do conhecimento [Looi et al. 2014, da Cruz Santos et al. 2024]. Isso resulta em um aumento do engajamento dos alunos no processo de aprendizagem. É importante destacar que a programação vai além do simples aprendizado da sintaxe de comandos; ela é uma atividade criativa que exige a capacidade de sequenciar instruções para resolver problemas.

A Olimpíada Brasileira de Informática (OBI) e a Maratona de Programação são competições científicas organizadas pela Sociedade Brasileira de Computação (SBC) com o objetivo de promover a criatividade, a capacidade de trabalho em equipe, a habilidade de resolver problemas e o domínio da programação. Essas competições representam um meio prático, democrático e poderoso para reunir estudantes para a resolução de problemas algorítmicos desafiadores, preparando-os para a construção de sistemas confiáveis e eficientes [Theodoro et al. 2024].

Em diversas universidades, existem grupos de estudo que incentivam a participação de alunos, a partir do primeiro semestre, nessas competições [Miranda et al. 2021]. No entanto, o levantamento estatístico da SBC revela que menos

de 8% dos estudantes de cursos na área de tecnologia participam da primeira fase da Maratona de Programação [Theodoro et al. 2024]. Consequentemente, uma grande parcela dos estudantes acaba não vivenciando os desafios e aprendizados proporcionados pelo formato de competições.

Dessa forma, uma abordagem local, inspirada no formato da Maratona de Programação e denominada MFUP, foi idealizada pelo Programa de Educação Tutorial (PET-SI) e desenvolvida para ser aplicada aos alunos da disciplina Fundamentos de Programação, ofertada no primeiro semestre dos cursos de graduação. Este artigo tem como objetivo detalhar a primeira aplicação desse projeto no curso de Sistemas de Informação da Universidade Federal do Ceará – Campus de Quixadá, cujos resultados apontam para um maior engajamento e motivação dos alunos, além de impactar positivamente no aprendizado da disciplina.

2. Fundamentação Teórica

Nesta seção, apresentamos a teoria que fundamenta o projeto MFUP.

2.1. Ensino de Programação em cursos de Computação

O ensino de programação nos cursos de Computação é conhecido por ser uma atividade desafiadora. Nesse sentido, [Souza et al. 2016] listam um conjunto de soluções para apoiar o ensino, as quais são classificadas nas seguintes treze categorias:

1. Visualização (S1): uso de animações para ilustrar a execução de programas.
2. Serious Games (S2): desenvolvimento e uso de jogos para ensino.
3. Ambientes de Desenvolvimento Pedagógico (S3): criação de ambientes para construção e execução de programas.
4. Colaboração (S4): estratégias para promover a aprendizagem entre alunos.
5. Scaffolding (S5): adaptação das tarefas conforme o nível dos alunos.
6. Notações (S6): ensino com linguagens e notações mais familiares.
7. Reflexão (S7): aprendizado baseado na análise de experiências anteriores.
8. Feedback (S8): fornecimento de retornos mais significativos sobre os programas dos alunos.
9. Instrução Ancorada (S9): resolução de problemas com base em material instrucional prévio.
10. Interatividade (S10): estímulo à interação entre alunos e professores.
11. Problemas Reais (S11): desenvolvimento de programas com aplicações práticas.
12. Representações Semânticas (S12): uso de representações mais naturais para compreensão de código.
13. Outros (S13).

A abordagem da **MFUP** está relacionada com várias dessas categorias como Colaboração (S4), Scaffolding (S5), Feedback (S8) e Problemas Reais (S11). Na próxima subseção, apresentaremos a estratégia utilizada pelo projeto **MFUP**, que é conhecida como **Aprendizagem por Descoberta**.

2.2. Aprendizagem por Descoberta

[Nikházy 2020] apresenta um método de aprendizagem por descoberta adaptado para o ensino de programação. O processo ocorre em um ciclo de cinco etapas: Problema, Investigação, Solução, Implementação e Análise, conforme ilustrado na Figura 1.

Figura 1. Processo da aprendizagem por descoberta no ensino de programação [Nikházy 2020]



- Problema – Os alunos analisam as informações e dicas disponíveis, tanto explícitas quanto implícitas, considerando também as restrições envolvidas.
- Investigação – Com apoio do professor, os alunos exploram o problema, decompondo-o e formulando perguntas orientadoras, como ”O que já sabemos sobre isso?”, além de esclarecer dúvidas sobre o enunciado.
- Solução – Antes da implementação, os alunos testam suas ideias para identificar e corrigir possíveis falhas na abordagem proposta.
- Implementação e Análise – Após a implementação, os alunos organizam suas reflexões, avaliam a solução e consolidam o aprendizado.

Acreditamos que esse método estimula a autonomia dos alunos, promovendo uma aprendizagem ativa e reflexiva, alinhada com as teorias construtivistas.

3. Trabalhos Relacionados

[Piekarski et al. 2015] relatam um projeto de extensão na UNICENTRO focado na preparação de alunos para a Maratona de Programação da SBC, utilizando encontros regulares, provas seletivas e competições simuladas. O estudo destaca desafios como a alta taxa de evasão e a dificuldade de conciliar o treinamento com as disciplinas curriculares.

[Miranda et al. 2021] descrevem a preparação de estudantes para a OBI Nível Sênior na UFC, incluindo o desenvolvimento da plataforma Code Marathon para auxiliar no treinamento por meio de materiais estruturados e simulados. O estudo reforça a importância das competições no aprendizado de algoritmos e estrutura de dados, mas aponta desafios como nivelamento de conhecimento e adesão dos alunos.

[Theodoro et al. 2024] analisam hábitos e estratégias de estudo de competidores de alto nível, destacando a importância da disciplina, planejamento de tempo e prática contínua em simuladores online. O estudo conclui que a produção e revisão do próprio material de consulta é mais relevante do que apenas tê-lo disponível na competição.

[de Sousa Escudero et al. 2023] apresentam um projeto de ensino para o desenvolvimento de habilidades em programação competitiva, utilizando encontros síncronos e grupos personalizados. Os resultados indicam que esse modelo melhora o engajamento e a capacitação dos alunos interessados em competições.

Enquanto os trabalhos anteriores focam na preparação para competições de alto nível, a MFUP adota uma abordagem inclusiva, voltada para alunos iniciantes em Fundamentos de Programação. Ao invés de exigir conhecimento prévio, a MFUP utiliza o formato de maratonas de programação como um método ativo de ensino, permitindo que os alunos aprendam colaborativamente e reforcem conceitos essenciais da disciplina.

4. O Projeto MFUP

O projeto da Maratona de Fundamentos de Programação (MFUP) foi estruturado para proporcionar uma experiência de aprendizado competitiva e interativa aos alunos matrículados na disciplina Fundamentos de Programação, usualmente ofertada no primeiro semestre dos cursos de graduação da área de tecnologia. Dessa forma, esta seção apresenta a metodologia proposta, que foi estruturada de maneira a possibilitar sua reutilização em outras universidades e será detalhada a seguir.

4.1. Metodologia

A Figura 2 apresenta uma visão geral das cinco etapas da MFUP. Após o professor da disciplina demonstrar interesse em aplicá-la com sua turma, a metodologia pode ser iniciada com a etapa 1, na qual sugerimos que o professor defina uma bonificação para os alunos participarem do projeto, como forma de motivar a inscrição de equipes.



Figura 2. Metodologia da MFUP

A Etapa 2 consistiu em iniciar a formação das equipes de alunos e a definição do dia e horário dos encontros, uma vez que o projeto ocorre extra sala de aula. Para isso, elaboramos um formulário eletrônico para coletar as seguintes informações: apelido da equipe, quantidade de participantes, nome e matrícula dos participantes, o dia da semana e o horário disponível para os encontros, o qual foi determinado pela opção mais escolhida no formulário. Recomendamos que a equipe seja formada entre 2 e 4 alunos, garantindo maior flexibilidade e alinhamento com o modelo da Maratona da SBC, que tradicionalmente conta com grupos de três participantes.

Uma vez que as equipes são definidas e o dia e horário dos encontros são estabelecidos, a Etapa 3 consistiu na realização dos encontros, os quais foram inspirados no método de aprendizagem por descoberta pensado, descrito na Seção 2. Cada encontro foi dividido em três fases principais: (i) resolução de problemas em equipe, estimulando a troca de conhecimento entre os integrantes; (ii) apresentação das soluções, permitindo que os alunos explicassem suas abordagens e estratégias de resolução; e (iii) discussão coletiva, onde dúvidas eram esclarecidas e novas técnicas eram introduzidas. Recomendamos a resolução de três problemas por encontro, considerando que o tempo total era limitado a duas horas e, em alguns casos, os alunos precisavam de mais tempo para encontrar a solução.

Durante os encontros, os participantes discutiam os problemas até que a primeira equipe encontrasse uma solução. A equipe que solucionasse o problema primeiro expli-

cava o caminho adotado para os demais, possibilitando que os outros alunos sugerissem melhorias ou compartilhassem as soluções que haviam pensado.

Os problemas foram elaborados com base nos conteúdos abordados em sala de aula, alinhando-os aos níveis e tipos de exercícios da Beecrowd, uma plataforma de juízes automáticos voltada para programação competitiva. Escolhemos a Beecrowd por sua facilidade de acesso, familiaridade dos alunos e suporte a torneios individuais e em equipe. Além de fornecer um vasto repositório de problemas, a plataforma oferece feedback automático, auxiliando no aprendizado e no desenvolvimento de habilidades em lógica, estrutura de dados e resolução de problemas.

Após os encontros, a Etapa 4 foi iniciada com uma competição final, que deve ser conduzida sem consulta a materiais externos, simulando uma maratona de programação real. Essa etapa foi, então, estruturada para ser composta por 12 questões, distribuídas em três níveis de dificuldade: fácil, médio e difícil. O total de questões foi definido com base no tempo disponível da competição, de modo a garantir que as equipes tivessem a oportunidade de resolver um número significativo de problemas sem comprometer a dinâmica do evento. A definição das questões seguiu os seguintes critérios:

- **Número de questões:** O total de 12 problemas foi estabelecido para equilibrar variedade e tempo disponível, permitindo que as equipes gerenciem sua estratégia e priorizem desafios conforme suas habilidades.
- **Níveis de dificuldade:** As questões foram categorizadas como fáceis, médias e difíceis, garantindo acessibilidade para alunos iniciantes, ao mesmo tempo que desafia participantes com maior domínio da programação.
- **Inspiração no Beecrowd:** Os problemas foram selecionados com base no repositório da plataforma Beecrowd, amplamente utilizada em competições de programação, permitindo que os alunos se familiarizassem com um ambiente de correção automática e testagem de código.
- **Cobertura de conceitos fundamentais:** As questões abrangeram temas essenciais da disciplina de Fundamentos de Programação, como estruturas condicionais, laços de repetição, manipulação de arrays e entrada/saída de dados.

Dessa forma, o conjunto de desafios foi estruturado para testar diferentes habilidades, promovendo tanto a aplicação prática dos conceitos aprendidos em sala quanto o desenvolvimento de estratégias de resolução sob pressão, respeitando o tempo disponível da competição.

Por fim, na Etapa 5, uma pesquisa anônima voluntária foi realizada para avaliar a satisfação com o projeto e, assim, melhorias serem propostas e aplicadas em uma próxima turma¹.

4.2. Primeira Execução da MFUP

A primeira aplicação do projeto foi em uma turma da disciplina de Fundamentos de Programação, do curso Sistemas de Informação da Universidade Federal do Ceará, Campus de Quixadá, no semestre 2024.1. A turma era composta por 49 alunos, com

¹O questionário elaborado para o projeto se encontra aqui: <https://bit.ly/perguntasfeedbackmfup>

carga horária de 92h na disciplina, ministrada na linguagem C. A MFUP iniciou no dia 13/08/2024 e finalizou no dia 17/09/2024.

O professor da turma estabeleceu que os alunos receberiam 0.5 na segunda avaliação, que era uma prova, por participação na competição e 1.0 para a equipe vencedora. Além disso, os alunos receberiam 10 horas de atividades complementares. Como o PET desenvolve um programa de acompanhamento de calouros — uma iniciativa que visa orientar e apoiar os estudantes ingressantes, ajudando-os a se ambientar na universidade por meio de consultorias informativas — essa iniciativa gerou um interesse ainda maior entre os estudantes, pois já estavam familiarizados com outras ações.

Tivemos a inscrição de 8 equipes com 4 alunos e 2 equipes com 3 alunos. No total, 38 alunos se interessaram em participar do projeto, de um total de 47 discentes da turma. A frequência da atividade ficou estipulada que seria realizada semanalmente, programada para ocorrer às terças-feiras, das 13h30 às 15h30, um horário viável para a maioria dos inscritos.

No total, foram realizados 5 encontros. A porcentagem de frequência dos estudantes foi superior a 76%. A plataforma utilizada foi o beecrowd acadêmico. A cada encontro as equipes tinham que resolver 10 problemas. No total, foram disponibilizados 30 problemas na plataforma para estudo dos estudantes².

Na simulação da maratona, as 10 equipes reuniram-se em um laboratório, sob a supervisão de 8 bolsistas do PET-SI, para resolver 10 questões. Ao final, duas equipes alcançaram um empate, com 7 problemas resolvidos cada. Por fim, a coleta de feedback realizada com o questionário contou com 13 respostas, totalizando 34% de participação. Os resultados são detalhados a seguir.

4.3. Resultados

A Figura 3 apresenta os resultados das questões quantitativas. Os dados apontam que 76% dos participantes manifestaram um alto grau de satisfação, enquanto 69% foram impactados significativamente no disciplina FUP, dos quais 3 participantes relatam impacto muito alto e 6 impacto alto.

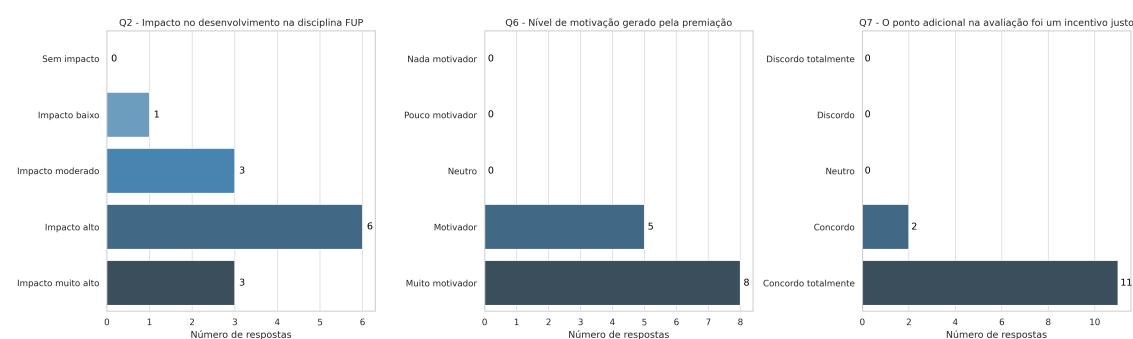


Figura 3. Resultados Quantitativos³

Revisão de Conteúdo e Discussão de Desafios e Dúvidas foram os pontos mais indicados como valiosos nos encontros, demonstrando evidências que a MFUP pode ser útil

²Os problemas se encontram aqui: <https://bit.ly/questoesmfup>

³Todos os outros resultados se encontram aqui: <https://bit.ly/graficoresultados>

para todos os alunos da disciplina. Adicionalmente, a maioria dos participantes concordaram que a duração do evento e o tempo de resolução dos problemas foram adequados.

Sobre a plataforma Beecrowd, uma grande parte gostou da escolha (38.5% - 5 participantes) e outra parte gostou muito (30.8% - 4 participantes). Além disso, 84.6% dos participantes indicaram que a bonificação foi justa e 61% dos respondentes identificaram a metodologia competitiva como um fator motivacional importante durante os encontros.

Com relação aos dados qualitativos, que correspondem com as perguntas 8, 9 e 10 do questionário, as respostas dos participantes foram analisadas e resumidas em duas categorias: (i) aspectos positivos e (ii) aspectos a serem melhorados.

Com relação aos aspectos positivos, a maioria dos participantes destacou pontos apreciados, tais como

- Revisão de conhecimentos: muitos participantes valorizaram a oportunidade de revisar conteúdos de FUP de maneira aplicada.
- Ambiente de competição saudável: a competição foi vista como um desafio motivador, mas sem pressão excessiva. Participante 6 relatou que o que mais gostou foi "*A competitividade de uma forma não 'tóxica'*".
- O campeonato final: o evento de encerramento foi considerado divertido e envolvente.
- As atividades propostas: os desafios e problemas apresentados ajudaram significativamente no aprendizado.
- A plataforma: um participante elogiou a plataforma utilizada, destacando que o projeto foi muito bom e que gostou muito de participar.
- O incentivo ao longo da maratona: A organização e o suporte ao longo do percurso foram bem recebidos.

Como aspectos a serem melhorados, os participantes destacaram os seguintes:

- Mais ensino teórico e explicações rápidas: alguns participantes sugeriram um reforço no ensino teórico e explicações sucintas sobre os tópicos abordados antes de cada sessão da maratona.
- Encontros on-line: a possibilidade de encontros on-line foi mencionada como uma forma de ampliar o acesso às discussões e revisões de conteúdo.
- Maior revisão dos conceitos de Fundamentos de Programação: foi sugerida uma revisão mais aprofundada sobre os tópicos essenciais da disciplina para reforçar o aprendizado.
- Equilíbrio entre questões de estudo e de competição: alguns participantes sugeriram um maior foco em questões voltadas para estudo, ao invés de tantas questões competitivas.
- Diversificação do conteúdo: inclusão de diferentes tipos de problemas e abordagens foi recomendada para atender a diversos estilos de aprendizado.
- Discussão aprofundada dos tópicos: foi sugerido um tempo extra para discutir e esclarecer dúvidas, beneficiando aqueles que aprendem em ritmos diferentes.
- Melhoria na plataforma utilizada: um relato apontou que a plataforma apresentou erros na correção, mesmo quando a lógica estava correta, indicando a necessidade de ajustes técnicos.
- Aumento da pontuação na disciplina: Um pedido recorrente foi a atribuição de mais pontos na disciplina como incentivo à participação.

5. Discussão e Lições Aprendidas

A primeira edição da MFUP proporcionou uma experiência inovadora para os alunos de Fundamentos de Programação, combinando aprendizado prático e gamificação. A análise dos dados coletados na pesquisa revelou uma recepção majoritariamente positiva, com destaque para o ambiente competitivo saudável, o impacto na motivação dos alunos e a relevância dos desafios propostos. No entanto, algumas sugestões de melhorias foram apontadas pelos participantes. Nesta seção, discutimos as principais lições aprendidas e as adaptações consideradas para futuras edições.

O Papel da MFUP: Competição vs. Ensino Teórico: alguns participantes sugeriram a inclusão de mais ensino teórico e explicações detalhadas sobre os tópicos abordados antes das competições. Esse feedback indica que, para parte dos alunos, a MFUP foi interpretada como uma extensão da monitoria da disciplina. No entanto, a proposta do projeto não é substituir aulas teóricas, mas proporcionar um espaço complementar para a aplicação prática de conceitos por meio de desafios competitivos. Diante disso, embora a MFUP não tenha o objetivo de se tornar uma atividade de ensino tradicional, podemos implementar ajustes que reforcem sua proposta pedagógica. Uma solução viável é adicionar revisões rápidas no início dos encontros, contextualizando os problemas do dia com base nos conteúdos já abordados em sala de aula.

Flexibilidade e Acessibilidade: outra sugestão é a realização de encontros online, visando maior flexibilidade para os participantes. Consideramos essa uma sugestão válida, especialmente para alunos que possuem restrições de horário ou dificuldades de deslocamento. Assim, uma alternativa para futuras edições é a implementação de encontros híbridos, permitindo que os alunos acompanhem remotamente ou revisitem discussões por meio de materiais gravados.

Equilíbrio entre Competição e Aprendizado: os dados qualitativos mostraram que alguns participantes sentiram a necessidade de um equilíbrio maior entre questões de estudo e competição. Como o caráter competitivo da MFUP é um dos seus diferenciais, acreditamos que esse equilíbrio pode ser atingido por meio de desafios progressivos. Para isso, nas próximas edições, os encontros poderão ser estruturados em três fases: (i) Problemas introdutórios para revisão de conceitos básicos; (ii) Problemas intermediários para reforço de aprendizado; e (iii) Desafios mais complexos no formato competitivo. Assim, a MFUP pode atender tanto alunos que buscam aprofundamento quanto aqueles que desejam um ambiente mais desafiador.

Diversificação do Conteúdo: outro ponto levantado foi a necessidade de incluir diferentes tipos de problemas e abordagens. Para isso, planeja-se ampliar o repertório de desafios, abordando (i) Desafios voltados para raciocínio lógico; e (ii) Questões com maior foco em algoritmos e estruturas de dados. Essa diversificação pode tornar a experiência ainda mais alinhada com a disciplina para incluir diferentes perfis de alunos.

Infraestrutura e Plataforma: alguns participantes relataram dificuldades com a correção automática da plataforma utilizada (Beecrowd). Embora seja uma ferramenta consolidada em competições de programação, será feita uma avaliação para identificar possíveis inconsistências. Caso os problemas persistam, consideraremos a implementação de um sistema complementar para validação de soluções ou até a migração para uma plataforma alternativa.

Incentivos e Gamificação: coletamos sugestões para aumentar a pontuação atribuída à participação na MFUP dentro da disciplina. Como a definição da pontuação cabe ao professor responsável, essa questão deve ser avaliada em conjunto com a coordenação do curso e o docente da disciplina para verificar a viabilidade de ajustes no incentivo acadêmico. Além disso, a MFUP já conta com premiações simbólicas, que foram bem recebidas pelos participantes. Para futuras edições, a organização pode explorar maneiras de tornar esses prêmios ainda mais atrativos, reforçando o aspecto motivacional da competição.

6. Considerações Finais e Próximos Passos

Esse artigo apresentou o projeto MFUP, idealizado pelo grupo PET-SI da Universidade Federal do Ceará, Campus de Quixadá. O projeto foi pensado para disciplinas de programação do primeiro semestre dos cursos de Computação e inspirado nos formatos de competição da Maratona de Programação. A MFUP consiste em formar equipes de alunos para competirem através da resolução de problemas.

Através da avaliação realizada, é perceptível que a MFUP teve um impacto positivo na experiência de aprendizado dos estudantes ingressantes. O projeto será executado novamente no primeiro semestre de 2025. Para essa futura edição, iremos adaptar o projeto conforme discutido na Seção 5.

Referências

- Bosse, Y. and Gerosa, M. A. (2017). Why is programming so difficult to learn? patterns of difficulties related to programming learning mid-stage. *ACM SIGSOFT Software Engineering Notes*, 41(6):1–6.
- da Cruz Santos, C., Araújo, R. D., and de Souza Pereira, J. H. (2024). Utilizando a abordagem de aprendizagem baseada em problemas para treinamento de programação competitiva com estudantes do ensino médio. In *Simpósio Brasileiro de Educação em Computação (EDUCOMP)*, pages 39–40. SBC.
- de Sousa, R. R. and Leite, F. T. (2020). Usando gamificação no ensino de programação introdutória. *Brazilian Journal of Development*, 6(6):33338–33356.
- de Sousa Escudero, P., Pasqua, Y. P., Jacó, A. P. G., de Oliveira, A. R., and Botelho, T. G. (2023). Desenvolvimento de estratégias de treinamento e monitoria para competição de programação. *16º JORNADA CIENTÍFICA E TECNOLÓGICA E 13º SIMPÓSIO DE PÓS-GRADUAÇÃO DO IFSULDEMINAS*, 15(2).
- Giraffa, L. M. M. and da Costa Mora, M. (2013). Evasão na disciplina de algoritmo e programação: Um estudo a partir dos fatores intervenientes na perspectiva do aluno. Acessado em: 19 mar. 2025.
- Looi, H. C., Seyal, A. H., and Darussalam, B. (2014). Problem-based learning: An analysis of its application to the teaching of programming. *International Proceedings of Economics Development and Research*, 70(14):68–75.
- Maciel, D. L., França, A. B., and Soares, J. M. (2013). Sistema de apoio a atividades de laboratório de programação via moodle com suporte ao balanceamento de carga e análise de similaridade de código. *Revista Brasileira de Informática na Educação*, 21(01):91.

- Miranda, P., Sousa, S., de Freitas Costa, J. R., Silva, G. I. O., de Souza Lima, V., Tavares, W. A., and Bezerra, C. I. M. (2021). Preparação para olimpíada brasileira de informática nível sênior: Um relato de experiência. In *Workshop sobre Educação em Computação (WEI)*, pages 101–110. SBC.
- Nikházy, L. (2020). Planets: A system for autonomous learning of algorithmic programming. In *ISSEP (CEURWS Volume)*, pages 79–90.
- Piekarski, A. E., Miazaki, M., Hild, T., Mulati, M. H., and Kikuti, D. (2015). A metodologia das maratonas de programação em um projeto de extensão: um relato de experiência. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 4, page 1246.
- Piteira, M. and Costa, C. (2013). Learning computer programming: study of difficulties in learning programming. In *Proceedings of the 2013 International Conference on Information Systems and Design of Communication*, pages 75–80.
- Raposo, E. H. S. and Dantas, V. (2016). O desafio da serpente-usando gamification para motivar alunos em uma disciplina introdutória de programação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 27, page 577.
- Souza, D. M., da Silva Batista, M. H., and Barbosa, E. F. (2016). Problemas e dificuldades no ensino de programação: Um mapeamento sistemático. *Revista Brasileira de Informática na Educação*, 24(1):39.
- Theodoro, L. C., da Cruz Santos, C., Irion, C., Araújo, R. D., de Souza Pereira, J. H., and Rosa, P. F. (2024). Compreendendo o sucesso em competições de programação: Perspectivas dos estudantes com resultados excepcionais. In *Workshop sobre Educação em Computação (WEI)*, pages 466–476. SBC.
- Venero, M. F. and Mena-Chalco, J. (2019). Ensino de programação avançada incentivando a metacognição: uma experiência positiva usando moodle+ vpl. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 30, page 279.