

Uma Proposta de Encontros de Tutoria Baseada em Metodologias Ativas para Disciplinas de Programação Introdutória

Laís de Paiva Freire¹, Jarbele Cássia da Silva Coutinho¹, Verônica Maria Lima Silva¹, Náthalee Cavalcante de Almeida Lima¹

¹Centro Multidisciplinar de Pau dos Ferros (CMPF)

Universidade Federal Rural do Semi-Árido (UFERSA)

BR 226, Km 405 – São Geraldo, Pau dos Ferros, RN | CEP: 59900-000

lais.freire@alunos.ufersa.edu.br, {jarbele.coutinho, veronica.lima, nathalee.almeida}@ufersa.edu.br

Abstract. *The introductory disciplines of programming in Computer courses present high rates of failure and dropout, in addition to difficulties related to the teaching methodology adopted, as well as other factors involved. Given this scenario, this paper presents a planning of sixteen mentoring meetings, based on active methodologies for introductory programming disciplines. In addition, an exploratory study was conducted to verify the acceptability of these meetings. In the obtained results, it was verified the relevance of the proposal as well as the strategies suggested for each meeting, according to the opinion of the students who participated in the exploratory study.*

Resumo. *As disciplinas introdutórias de programação nos cursos de Computação possuem elevados índices de reprovação e evasão, bem como dificuldades ligadas à metodologia de ensino adotada, além de outros fatores envolvidos. Diante desse cenário, este trabalho apresenta um planejamento de dezesseis encontros de tutoria, baseados em metodologias ativas para disciplinas de programação introdutória. Além disso, realizou-se um estudo exploratório a fim de verificar a aceitabilidade desses encontros. Nos resultados obtidos, foi constatada a relevância da proposta bem como das estratégias sugeridas para cada encontro, de acordo com a opinião dos alunos que participaram do estudo exploratório.*

1. Introdução

Disciplinas de programação exigem do aluno conhecimentos prévios em lógica matemática, leitura e interpretação de texto, abstração de ideias, dentre outras habilidades. Entretanto, boa parte dos alunos apresentam dificuldades em desenvolver tais aptidões. Na literatura são apresentadas diversas justificativas para as dificuldades próprias ao processo de ensino-aprendizagem de programação [Ambrósio e Costa, 2010; Carvalho, Maia e Cavalcante, 2017; Mourão, 2017].

Ambrósio e Costa (2010) defendem que a aprendizagem de programação deve ser entendida como um processo lento e gradual, e que um dos principais fatores que influenciam neste processo é o ritmo de aprendizagem do aluno. Esta diversidade de

ritmos de aprendizagem, aliada, na maioria dos casos, a turmas muito numerosas, constitui um cenário pouco investigado, mas crucial ao desenvolvimento das habilidades e ao aprendizado de programação [Holanda, Coutinho e Fontes, 2018]. Além disso, aliar teoria com atividades práticas bem como prover acompanhamentos individuais, em turmas numerosas, não é uma tarefa trivial para os professores que conduzem tais disciplinas [Carvalho, Maia e Cavalcante, 2017].

Questiona-se, portanto: como conduzir e acompanhar o aprendizado do aluno de programação introdutória, de forma individualizada, em turmas numerosas, respeitando o ritmo de aprendizagem e estimulando a autonomia e participação de cada um? Desse modo, a partir desta problemática percebeu-se a necessidade de oferecer ao aluno de programação introdutória um acompanhamento mais direcionado às suas dificuldades, respeitando a individualidade e o ritmo de seu aprendizado.

Assim, este trabalho apresenta um planejamento de dezesseis encontros de tutoria, baseado em metodologias ativas, como forma de proporcionar ao aluno um aprendizado individualizado, autônomo e participativo. Para isto, a proposta dos encontros de tutoria baseou-se no Programa Geral do Componente Curricular (PGCC) da disciplina de Algoritmos de um curso de graduação em Tecnologia da Informação da Universidade Federal Rural do Semi-Árido, Centro Multidisciplinar de Pau dos Ferros, e em resultados de um estudo exploratório realizado com alunos de uma turma de Algoritmos.

Espera-se, portanto, contribuir também com a mitigação dos principais desafios enfrentados na aprendizagem de programação introdutória, apontados por Moreira e Holanda (2018), como: dificuldade de raciocínio lógico, compreensão do enunciado das questões, entendimento da sintaxe da linguagem de programação, dificuldade de concentração e aulas repetitivas (metodologicamente).

Por fim, o presente trabalho encontra-se organizado da seguinte forma: a Seção 2 discute o problema abordado neste estudo; a Seção 3 apresenta trabalhos correlatos; a Seção 4 descreve os procedimentos metodológicos adotados; a Seção 5 discute os principais resultados obtidos com o estudo exploratório e apresenta a proposta dos encontros de tutoria; e, por fim, a Seção 6 traz as considerações finais.

2. Tutoria no Ensino de Programação Introdutória

Comumente, a trajetória a ser percorrida por um aluno em um curso da área da Computação inicia em disciplinas de programação introdutória, na qual acontece o primeiro contato do aluno com: a abstração de informações para resolução de problemas; e, o aprendizado de uma linguagem de programação específica. Esses conhecimentos iniciais estimularão o desenvolvimento das habilidades e competências necessárias para todo o restante do curso.

Por isso, atualmente, diversas estratégias alternativas – como, metodologias, técnicas e ambientes educacionais - estão sendo investigadas e utilizadas para que o ensino-aprendizagem de programação se torne mais eficiente [Holanda, Coutinho e Fontes, 2018; Lopes et al., 2017; Santiago e Kronbauer, 2017; Soares e Carvalho, 2017; Holanda, Freire e Coutinho, 2019].

Semelhante ao que acontece na Educação à Distância (EaD), um sistema de tutoria é caracterizado por uma ou mais pessoas encarregadas de gerenciar um grupo de alunos matriculados em um componente curricular e oferecer um suporte individualizado para o aprendizado deste componente. Para [Scaico et al., 2012], uma atividade de tutoria se refere ao aconselhamento ou treinamento de um indivíduo a partir de alguém mais experiente, para que este consiga assimilar um conteúdo e possa transformar informações em conhecimento. Um tutor oferece atenção personalizada, diferentemente de um monitor que atua em um contexto de acompanhamento coletivo.

Experiências de tutoria em programação reforçam que, para que o trabalho seja eficiente, o tutor necessita compreender a realidade do aluno, adequar-se à sua velocidade de absorção de informação, bem como seu ritmo de aprendizado [Scaico et al., 2012]. Em contrapartida, é imprescindível que a temática a ser abordada na tutoria seja previamente acessada pelo aluno. Deste modo, a tutoria em programação é responsável pelo acompanhamento individualizado ao aluno, mas não pelo ensino prévio do conteúdo.

[Scaico et al., 2012] defende a importância da tutoria em programação, sendo esta responsável, também, por fornecer um apoio direcionado aos estudantes matriculados em disciplinas de programação introdutória. Dessa forma, caracteriza-se como um dos fatores mais importantes para seus respectivos resultados de aprendizagem, onde “as questões relacionadas à influência dos hábitos escolares do estudante nesse processo ainda é tratada com negligência” [Scaico et al., 2012].

Para isso, faz-se necessária a adoção de abordagens que estimulem a participação ativa e autônoma do aluno. Desse modo, as metodologias ativas vêm apoiar a inserção de práticas que estimulam o engajamento do aluno na tutoria. Para [Mourão, 2017], metodologias ativas de ensino são propostas voltadas para a participação ativa dos indivíduos de acordo com sua realidade, ou seja, tornam os alunos sujeitos ativamente envolvidos em seu próprio aprendizado. De modo complementar, [Paiva et al., 2016] destaca que metodologias ativas têm como principal característica a inserção do aluno como figura central do processo de ensino, tornando-o o principal responsável pela sua aprendizagem.

3. Trabalhos relacionados

No intuito de compreender a utilização de metodologias ativas no ensino de programação e em projetos de tutoria com alunos de programação introdutória, foram analisados alguns trabalhos relacionados.

Em Scaico et al. (2012) é apresentado um relato de experiência acerca da utilização de um modelo de tutoria aplicada à disciplina de Introdução à Programação da Universidade Federal da Paraíba. Como resultados, constatou-se que os índices de evasão na disciplina foram reduzidos em relação aos semestres anteriores, além do aumento da taxa de aprovação dos alunos – em que 40% dos alunos tutelados obtiveram êxito na disciplina, conseguindo aprovação por média.

Em Paiva et al. (2016) é apresentada uma revisão integrativa sobre metodologias ativas de ensino-aprendizagem nos cenários de educação, de modo a investigar os benefícios e desafios do seu uso. Os resultados apontam que a utilização de

metodologias ativas está presente desde a educação básica ao ensino superior. Em relação aos benefícios, destaca-se o desenvolvimento da autonomia do estudante. E, em relação aos desafios, verificou-se que a adaptabilidade do aluno bem como a formação profissional do educador são fortes empecilhos para a adoção dessas práticas.

Em Hauck et al. (2018) é apresentado o projeto Jovens Tutores de Programação, cujo objetivo consiste na capacitação de jovens dos últimos anos do ensino fundamental, de uma escola pública municipal de Florianópolis, através da programação de dispositivos móveis. Durante o projeto foram capacitados um total de três alunos tutores que ministraram a oficina de desenvolvimento *mobile*, para uma turma de cinquenta alunos. Como estudos futuros, o trabalho sugere a promoção de novas oficinas em outras escolas, utilizando a ferramenta *CodeMaster* desenvolvida pela Iniciativa Computação na Escola.

Assim, o trabalho descrito nesse artigo inspirou-se na importância de oferecer aos alunos novatos em programação um acompanhamento personalizado, através da junção de estratégias usadas na tutoria com a adoção de práticas de metodologias ativas, diferindo dos trabalhos relacionados apresentados anteriormente, onde as atividades eram planejadas de forma interativa a partir do contato dos tutores e seus tutelados conforme apresentado em [Scaico et. al., 2012]. Assim, as atividades presentes em cada um dos encontros de tutoria foram sugeridas a partir de um levantamento realizado com os alunos de uma turma de programação introdutória, de acordo com suas preferências, e de forma que a aplicação desse planejamento possa acontecer durante todo o semestre, diferente da abordagem adotada em [Hauck et. al., 2018].

4. Metodologia Adotada para Elaboração dos Encontros

Inicialmente, para planejar os encontros de tutoria, se fez necessário conhecer o conteúdo de uma disciplina de programação introdutória. Para isso, foi adotado, como base, o Programa Geral do Componente Curricular (PGCC) da disciplina de Algoritmos, inserida no primeiro semestre de um curso de graduação em Tecnologia da Informação da Universidade Federal Rural do Semi-Árido, Centro Multidisciplinar de Pau dos Ferros. Assim sendo, os encontros da tutoria foram distribuídos em dezesseis (16) momentos, de acordo com os conteúdos apresentados no referido PGCC. As atividades programadas para cada encontro de tutoria foram idealizadas a partir das práticas de metodologia ativa Baseada em Problemas (*Problem Based Learning* - PBL). Para cada encontro foi proposto um plano de atividades, baseado na seguinte estrutura:

- *Identificação do encontro*: conteúdo principal a ser discutido;
- *Identificação do tutor*: nome do tutor responsável pela realização do encontro;
- *Data/Local*: data e local onde o encontro será realizado;
- *Descrição*: descreve o objetivo geral do encontro;
- *Contribuições*: descreve os principais benefícios que o encontro trará ao aluno;
- *Recursos a serem utilizados*: descreve os recursos necessários à realização do encontro, tais como: lousa, lápis, papel, computador, etc;
- *Conteúdo abordado*: descreve o conteúdo a ser abordado no encontro;
- *Metodologia*: descreve a metodologia adotada, ou seja, o modo como as atividades serão conduzidas no encontro;
- *Requisitos necessários*: descreve as pré-condições necessárias para os alunos

que desejam participar dos encontros;

- *Sugestão de softwares*: descreve uma lista de *softwares* adicionais que irão auxiliar o aprendizado e que estão relacionados ao conteúdo do encontro;
- *Recursos complementares*: descreve uma lista de outros recursos complementares às atividades realizadas no encontro de tutoria (por exemplo: questões relacionadas, jogos educativos, referências à materiais didáticos, entre outros).

De modo complementar e com a finalidade de obter um *feedback* inicial acerca das atividades que foram sugeridas no planejamento dos encontros de tutoria foi realizado um estudo exploratório com uma amostra de dezessete (17) alunos de uma turma de programação introdutória, através de um formulário *online*, composto por seis (06) questões objetivas e uma (01) questão de múltipla escolha (ver Quadro 1).

Quadro 1. Questões abordadas no formulário

| Pergunta | Possível resposta | Pergunta | Possível resposta |
|--|---|---|---|
| Você já participou de um projeto de tutoria? | Sim Não | Você conhece como funciona um projeto de tutoria? | Sim Não |
| Você gostaria de participar de um projeto de tutoria na disciplina de Algoritmos? | Sim Não | Para você, qual a relevância da presença de um tutor na disciplina de Algoritmos | Relevante Irrelevante |
| Você gostaria de ter um tutor na disciplina de Algoritmos? | Sim Não | Ter participado de um projeto de tutoria teria feito diferença nas suas notas na disciplina de Algoritmos? | Sim Não |
| O que você gostaria que fosse abordado em um projeto de tutoria na disciplina de Algoritmos? | Uso de jogos Revisão de conteúdo Resolução de questões Simulação de prova Maratona de programação | Em uma escala de 0 a 5, sendo 0 irrelevante e 5 muito significativo, classifique o impacto da tutoria no seu processo de aprendizagem na disciplina de Algoritmos | 0 1 2 3 4 5 Não participei do projeto |

5. Discussão e Resultados

Apresentamos e discutimos a seguir os resultados obtidos com o estudo exploratório realizado a fim de validar as atividades a serem realizadas dentro da tutoria e, por conseguinte expomos o planejamento dos encontros propostos.

5.1. Resultados do Estudo Exploratório

Os dados obtidos através do estudo exploratório foram sintetizados e analisados. Observou-se, portanto, que 88% dos alunos (ou 15 deles) nunca participaram de atividades de tutoria. Enquanto que 12% deles, afirmaram já ter participado de atividades de acompanhamento individualizado.

Em relação ao conhecimento dos alunos sobre o funcionamento de atividades de tutoria, observou-se que 41% deles (equivalente a 07 alunos) não conhecem como funcionam as atividades de tutoria, enquanto que os outros 59% (equivalente a 10 alunos) afirmaram ter conhecimento sobre tais atividades.

Foi questionado, também, sobre o interesse dos alunos em participar de atividades de tutoria durante o curso de uma disciplina de Algoritmos e observou-se que, a maioria dos alunos (equivalente à 88% ou 15 deles) afirmaram possuir interesse nesta ação. Enquanto que, uma minoria, equivalente 12% dos alunos, afirmou não possuir interesses em atividades dessa categoria.

Em relação às abordagens a serem adotadas nas atividades de tutoria (consultar Gráfico 01), os alunos apontaram uma preferência pela realização de maratonas de programação (76% ou 13 alunos), uso de jogos (58% ou 10 alunos), atividades de revisão de conteúdo (47% ou 08 alunos), simulações de avaliação (41% ou 07 alunos), e auxílio na resolução de questões (41% ou 07 alunos).

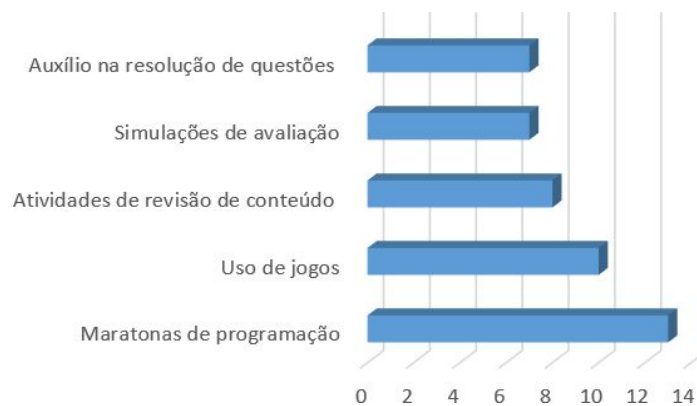


Gráfico 1. Propostas de atividades informadas pelos alunos

Em relação ao acompanhamento personalizado durante a disciplina, 94% dos alunos (ou 16 deles) afirmaram ter interesse em receber este acompanhamento. Enquanto que 06% deles, afirmaram não possuir interesse. Os mesmos 94% dos alunos consideraram relevante o acompanhamento individualizado durante a disciplina de programação.

Por fim, percebeu-se que 88% dos alunos (equivalente a 15) afirmaram que um acompanhamento personalizado durante a disciplina contribuiria consideravelmente com o seu aprendizado e, conseqüentemente, com um melhor desempenho na disciplina de programação. Enquanto que outros 12% (equivalente a 02 alunos), afirmaram não perceber diferença nessa contribuição.

De um modo geral, foi percebida a importância da presença de um tutor e de sua ação personalizada às necessidades do aluno de programação. Foram também consideradas sugestões de atividades que contemplassem as propostas apontadas pelos alunos, a citar: (i) as opções de revisão do conteúdo visto em sala de aula são ofertadas, nos encontros de tutoria, em forma de sugestão de vídeo-aulas e interação com jogos educativos que englobam o conteúdo; (ii) as resolução de questões, nos encontros de tutoria, são indicadas através ferramentas (como, o URI Online Judge) que promovem a prática de programação, oferecendo *feedback* imediato acerca das resoluções, além da opção de promover uma maratona de programação entre os alunos. A partir das opiniões coletadas, foi possível validar a possível aceitação das atividades propostas nos encontros, de modo que esses possam vir a ser aplicados posteriormente.

5.2. Descrição Geral dos Encontros da Tutoria

A comunicação entre o tutor-gerente, o professor da disciplina e o coordenador do projeto deve ocorrer em reuniões mensais, para que informações coletadas pelos tutores, referentes aos tutorados, sejam reportadas e discutidas. Reuniões semanais entre o

tutor-gerente e os demais tutores serão realizadas a fim de socializar os resultados observados durante cada encontro.

Os encontros de tutoria devem ser conduzidos por alunos (tutores) que já cursaram uma disciplina de programação introdutória. Para a realização dos encontros, têm-se os seguintes papéis, conforme ilustrado na Figura 1:

- *Tutor-gerente*: aluno responsável pelo gerenciamento das atividades dos demais tutores e que tem a incumbência de realizar a comunicação entre os coordenadores do projeto e os demais membros da equipe, não atuando diretamente nos encontros da tutoria;
- *Tutores*: alunos que já cursaram a disciplina de programação introdutória, e que serão responsáveis pelo acompanhamento semanal dos alunos, através dos encontros, atentando para o desempenho e participação dos alunos;
- *Tutorados*: alunos devidamente matriculados na disciplina de programação introdutória, que está sendo acompanhada pela equipe de tutoria;
- *Professor da disciplina*: professor responsável pela turma de alunos, assistida pelos tutores, e que colabora com as atividades de tutoria - por meio de informações da turma de alunos e das aulas ministradas;
- *Coordenador*: professor responsável pela equipe de tutoria.

A comunicação entre o tutor-gerente, o professor da disciplina e o coordenador do projeto deve ocorrer em reuniões mensais, para que informações coletadas pelos tutores, referentes aos tutorados, sejam reportadas e discutidas. Reuniões semanais entre o tutor-gerente e os demais tutores serão realizadas a fim de socializar os resultados observados durante cada encontro.

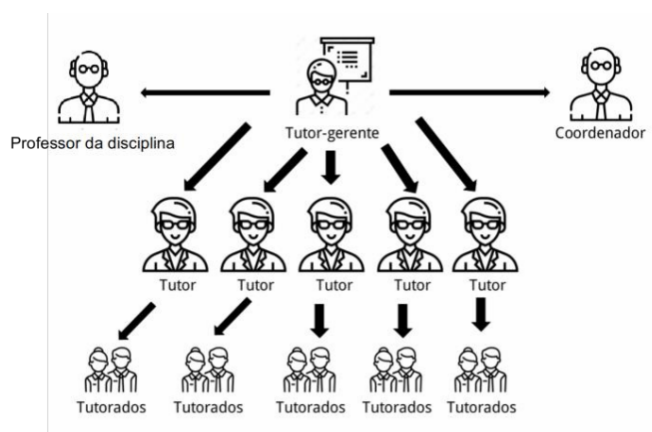


Figura 1. Papéis da equipe de tutoria

Para o desenvolvimento dos encontros de tutoria, a turma deve ser dividida de forma igualitária entre os tutores. A partir disso, cada tutor será responsável por realizar a comunicação entre seus tutorados, informando local e horário disponíveis para o atendimento. Sugere-se que a quantidade de tutorados por encontro seja entre um e três alunos, de forma que o tutor possa transitar tranquilamente entre seus tutorados e preservar a essência da tutoria: o encontro individual.

Desse modo, é proposto que os encontros de tutoria sejam executados semanalmente, totalizando uma série de dezesseis (16) encontros. No primeiro encontro, orienta-se que seja realizado um pré-teste com os alunos, a fim de avaliar suas

competências e habilidades em programação. Após isso, os encontros devem acontecer em paralelo com as aulas ministradas pelo professor da disciplina, de modo a fornecer um apoio ao aprendizado do aluno. Ao final dos encontros de tutoria, propõe-se a realização de uma pequena competição entre os tutorados, denominada “mini olimpíada de programação”, além da aplicação de um formulário avaliativo para verificar a evolução da aprendizagem dos alunos durante os encontros. O Quadro 2 apresenta uma síntese geral dos encontros de tutoria propostos.

Quadro 2. Síntese geral dos encontros de tutoria

| | Conteúdo abordado | Descrição do conteúdo |
|--------------|--------------------------|--|
| 1º Encontro | Pré-teste | Medir as aptidões do aluno. |
| 2º Encontro | Fluxogramas | Funcionamento, componentes e utilização. |
| 3º Encontro | Pseudocódigo | Estruturas de condição |
| 4º Encontro | Pseudocódigo | Auxiliar na resolução dos exercícios |
| 5º Encontro | Simulação de prova | Simular uma avaliação do conteúdo |
| 6º Encontro | Linguagem de programação | Estruturas de condição |
| 7º Encontro | Linguagem de programação | Estruturas de repetição <i>while</i> e <i>do-while</i> |
| 8º Encontro | Linguagem de programação | Estruturas de repetição <i>for</i> |
| 9º Encontro | Simulação de prova | Simular uma avaliação do conteúdo |
| 10º Encontro | Funções | Estruturar algoritmos com funções |
| 11º Encontro | Vetores | Declaração e inicialização |
| 12º Encontro | Vetores | Manipulação |
| 13º Encontro | Matrizes | Declaração e inicialização |
| 14º Encontro | Matrizes | Manipulação |
| 15º Encontro | Miniolimpíada | Competição entre o tutorados |
| 16º Encontro | Formulário de avaliação | Avaliação dos encontros |

É válido ressaltar que os encontros foram planejados de acordo com o PGCC de uma disciplina de Algoritmos, inserida no primeiro semestre de um curso de graduação em Tecnologia da Informação da Universidade Federal Rural do Semi-Árido, Centro Multidisciplinar de Pau dos Ferros, e que possuem a duração máxima de 90 minutos¹.

Por fim, o planejamento dos encontros se limitam ao PGCC da disciplina supracitada, mas podem ser replicados e alterados de acordo com o conteúdo programático da disciplina de programação que se queira abordar.

¹ Para acessar o planejamento personalizado dos encontros de tutoria, acesse: <<https://drive.google.com/open?id=1GF4JukdgrC0kegty39sMIV3k3npoMxC9>>

6. Considerações Finais

Neste trabalho é apresentado um planejamento de dezesseis encontros de tutoria, baseado em metodologias ativas, de forma a conduzir e acompanhar o aprendizado do aluno de programação introdutória, de maneira individualizada. Foi realizado, portanto, um estudo exploratório com alunos de programação para coletar informações sobre a relevância de um acompanhamento personalizado (a tutoria), a análise do PGCC de uma disciplina de programação introdutória e o estudo de práticas de metodologias ativas para serem inseridas como atividades, em cada encontro de tutoria.

Após o estudo exploratório, constatou-se um *feedback* positivo dos alunos em relação ao acompanhamento dos estudos através da tutoria. Os resultados obtidos neste estudo contribuíram, também, para o planejamento dos encontros de tutoria, tendo em vista a listagem de atividades atribuídas a cada encontro. Portanto, através dos encontros de tutoria, espera-se contribuir também com a mitigação dos principais desafios enfrentados na aprendizagem de programação introdutória.

Com relação às possíveis dificuldades a serem enfrentadas, pode-se destacar a provável incompatibilidade de horários entre o tutor e seus tutorados, além da própria presença dos alunos nos encontros de tutoria, uma vez que as disciplinas de programação introdutória estão inseridas nos primeiros períodos dos cursos de Computação. Assim, muitas vezes o aluno ainda não possui a maturidade necessária para compreender o impacto que intervenções dessa natureza possui.

Como trabalhos futuros pretende-se aplicar os encontros de tutoria utilizando os planejamentos elaborados, através do acompanhamento de alunos de uma turma de programação introdutória. Para tanto, um estudo experimental inicial se faz necessário para validar a eficácia da proposta, e para eventuais melhorias nas atividades de cada encontro proposto.

Referências

- Ambrósio, Ana Paula; Costa, Fábio Moreira. O uso de PBL para o ensino de Algoritmos e Programação de Computadores. Anais do Congresso Internacional - Pbl 2010, São Paulo, p.1-11, fev. 2010.
- Carvalho, Rodolfo; Maia, Dennys; Cavalcante, Weksley. Desenvolvimento de um objeto de aprendizagem em uma disciplina da graduação do curso Bacharelado de Tecnologia da Informação: um relato de experiência. Anais dos Workshops do VI Congresso Brasileiro de Informática na Educação (cbie 2017), [s.l.], p.56-60, 27 out. 2017. Brazilian Computer Society (Sociedade Brasileira de Computação - SBC).
- Grotta, Alexandre; Prado, Edmir P. V.. Um ensaio sobre a experiência educacional na programação de computadores: a abordagem tradicional versus a aprendizagem baseada em projetos. Anais do XXVI Workshop Sobre Educação em Computação (WEI), Natal, p.191-200, jul. 2018.

- Hauck, Jean Carlo Rossa et al. Jovens tutores de programação: um relato de experiência. *Extensio: Revista Eletrônica de Extensão*, [s.l.], v. 15, n. 29, p.94-108, 24 ago. 2018. Universidade Federal de Santa Catarina (UFSC).
- Holanda, Wallace; Coutinho, Jarbele; Fontes, Laysa. Uma Intervenção Metodológica para Auxiliar a Aprendizagem de Programação Introdutória: um estudo experimental. *Anais dos Workshops do VII Congresso Brasileiro de Informática na Educação (CBIE 2018)*, [s.l.], p.699-708, 28 out. 2018. Brazilian Computer Society (Sociedade Brasileira de Computação - SBC).
- Holanda, Wallace Duarte de; Freire, Laís de Paiva; Coutinho, Jarbele Cássia da Silva. Estratégias de ensino-aprendizagem de programação introdutória no ensino superior: uma Revisão Sistemática da Literatura. *Renote*, [s.l.], v. 17, n. 1, p.527-536, 28 jul. 2019. Universidade Federal do Rio Grande do Sul.
- Moreira, Gabriel Luídy; Holanda, Wallace. Desafios na aprendizagem de programação introdutória em cursos de TI da UFERSA, campus Pau dos Ferros: um estudo exploratório. *Anais do Encontro de Computação do Oeste Potiguar Ecop/UFERSA, Pau dos Ferros*, v. 2, n. 1, p.90-96, 21 mar. 2018.
- Mourão, Andreza. Uma proposta da eficiência do uso da Metodologia Ativa Baseada em Problemas, utilizando Dojo de Programação, aplicada na disciplina de Lógica de Programação. *Anais do XXIII Workshop de Informática na Escola (wie 2017ca)*, [s.l.], p.667-676, 27 out. 2017. Brazilian Computer Society (Sociedade Brasileira de Computação - SBC). <http://dx.doi.org/10.5753/cbie.wie.2017.667>.
- Paiva, Marlla Rúbya Ferreira et al. Metodologias ativas de ensino-aprendizagem: uma revisão integrativa. *Sanare: Revista de Políticas Públicas, Sobral*, v. 15, n. 2, p.145-153, dez. 2016.
- Ramos, Cristiane Soares et al. TBL as an active learning-teaching methodology for Software Engineering courses. *Anais do IX Congresso Brasileiro de Engenharia de Software, São Carlos*, p.289-297, set. 2018.
- Sáez-López, José-Manuel; Román-González, Marcos; Vázquez-Cano, Esteban. Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers & Education*, [s.l.], v. 97, p.129-141, jun. 2016. Elsevier BV.
- Scaico, Pasqueline et al. Relato de um modelo de tutoria para programação baseado em experiências com ingressantes de um curso de Licenciatura em Computação. *Anais dos Workshops do XX Congresso da Sociedade Brasileira de Computação (SCBS 2012)*, Curitiba, jul. 2012. Brazilian Computer Society (Sociedade Brasileira de Computação - SBC).
- Santiago, A. D. V.; Kronbauer, A. H. Um Modelo Lúdico para o Ensino de Conceitos de Programação de Computadores. In *Revista Brasileira de Informática na Educação*. v. 25, n. 3, p. 1-29. 2017.
- Soares, F. A. L.; Carvalho, R. B. Proposta de um Portal Educacional para estudantes de programação de computadores. In *Revista Abakós*. v. 5, n. 2, p. 36-58. 2017.