

# Uma Proposta de Boas Práticas para o Ensino Colaborativo de Programação de Computadores

Aline Zanin<sup>1</sup>, Adalto Selau Sparremberger<sup>2,3</sup>, Jorge L. V. Barbosa<sup>4</sup>

<sup>1</sup>Programa de Pós-Graduação em Ciência da Computação (PPGCC) – PUCRS

<sup>2</sup>Escola de Tecnologia da Informação – Centro Universitário FADERGS

<sup>3</sup>Escola de Tecnologia da Informação – Faculdade Senac Porto Alegre (FSPOA)

<sup>4</sup>Programa de Pós-Graduação em Computação Aplicada (PPGCA) – UNISINOS

aline.zanin@acad.pucrs.br, adaltoss@me.com, jbarbosa@unisinios.br

**Resumo.** *A programação de computadores é uma atividade complexa, que envolve saberes associados a diversas áreas de conhecimento. Por isso, os professores desta área precisam estar continuamente, desenvolvendo formas de ensinar e ajudar os alunos a reduzir a dificuldade no aprendizado. Este trabalho realizou uma survey, em formato de questionário, que foi enviada para programadores, visando entender como estes aprenderam a programar, quais as formas de ensino que mais auxiliaram neste processo e o que eles mudariam na forma como foram ensinados a programar. Este trabalho apresenta uma proposta de sugestões para apoio aos professores na tarefa de estruturação das aulas de programação. Esta proposta considera o ensino colaborativo de programação através de ferramentas adequadas para a aplicação de cada uma das práticas propostas.*

**Palavras-Chave:** ensino, colaborativo, programação, práticas

A Proposal for Good Practices for the Collaborative Teaching of Computer Programming

**Abstract.** *Computer programming is a complex activity that involves knowledge associated with various areas of knowledge. Therefore, teachers in this area need to be continuously developing ways to teach and help students reduce their learning difficulties. This work carried out a survey, in a questionnaire format, which was sent to programmers, in order to understand how they learned to program, which forms of teaching helped most in this process and what they would change in the way they were taught to program. This work presents a guide of suggestions to support teachers in the task of structuring the programming lessons. This guide considers the collaborative teaching of programming through appropriate tools for the application of each of the proposed practices.*

**Keywords:** teaching, collaborative, programming, practices

## 1. Introdução

O ensino de programação de computadores é uma atividade desafiadora para os docentes, que assumem a missão de construir junto aos estudantes, as competências que tangem esta área do conhecimento (Ferreira and Reategui 2016). Os desafios que emergem neste processo estão associados aos mais diversos fatores, por exemplo: dificuldade de entendimento do que é esperado num enunciado de um exercício, a escassez de tempo para realização de atividades e a falta de motivação oriunda de um ambiente escolar que não privilegia o diálogo e a colaboração. (Giraffa and Mora 2013).

Encontrar dificuldades, ou não ver evoluções no processo de aprendizagem são fatores que levam os estudantes a evadir dos cursos da área de programação (Gomes et al. 2008), sendo que, o curso de ciência da computação está na lista dos cursos com maior evasão em nível superior, tanto em universidades públicas quanto em universidades privadas (CBSI 2019). O fato desta evasão estar relacionada à dificuldade de aprendizagem é um fator que demanda atenção dos pesquisadores da área de educação. Esta atenção é necessária, especialmente porque o mercado de trabalho cresce continuamente em número de oportunidades para programadores. Atualmente, apenas para programadores da linguagem Java existem 949 vagas em aberto no Brasil em um *site* popular de oferta de empregos, o Infojobs (Infojobs 2019). Neste sentido, os educadores têm o desafio de entender como podem auxiliar os alunos no processo de aprendizagem, para que as dificuldades encontradas não sejam superiores ao desejo de concluir o curso.

Para auxiliar os alunos a superar as dificuldades e motivá-los no processo de aprendizagem, as estratégias de ensino utilizadas pelos professores são um fator determinante. Esta importância justifica-se uma vez que, quando o professor entende a forma como o aluno aprende (lendo, escrevendo, ouvindo, ou fazendo), ele pode preparar suas aulas utilizando meios de ensino que explorem o método mais adequado para a turma.

Em trabalhos anteriores, foi abordada a identificação do perfil dos estudantes em relação a proatividade, reatividade e interesse de aprendizes em ambientes virtuais de aprendizagem (Sparremberger et al. 2017), além do uso da *gamificação* como estratégia de ensino (Zanin et al. 2018). Em ambos os trabalhos identificou-se que conhecer o aluno e adotar estratégias de ensino adequadas ao perfil destes, estimula o aprendizado e melhora o envolvimento dos alunos com o curso para, por exemplo, realizarem atividades extraclasse com maior motivação. Sob a mesma perspectiva de potencializar o aprendizado do aluno, é sabido que, “a colaboração tem impacto determinante na construção do conhecimento pois, envolve níveis de cognição mais elaborados do que os envolvidos na ação individual de aprendizagem” (Castro and Menezes 2011). Neste sentido, se faz necessário concentrar esforços em desenvolver metodologias colaborativas para o ensino de programação. Estas metodologias precisam ter essência colaborativa por parte do professor que compartilha conhecimentos e materiais didáticos com alunos e colegas, e também entre os próprios estudantes que interagem entre si, para melhorar o desempenho individual de cada um.

Visando propor um conjunto de sugestões para o ensino de programação que considere o uso de ferramentas colaborativas e também o perfil de aprendizado dos estudantes de programação, efetuou-se uma revisão de literatura em formato de Snowballing (Wohlin 2014), partindo de artigos publicados no Simpósio Brasileiro de Informática na Educação (SBIE). Nesta revisão de literatura, não foram encontrados trabalhos que abrangessem especificamente, o ensino colaborativo de programação baseado nos métodos de ensino, que os programadores consideraram mais eficientes em seu processo de aprendizagem.

Para atender a este desafio, neste trabalho buscou-se entender, qual o ponto de vista de programadores a respeito das metodologias utilizadas durante o seu processo de formação. Neste contexto, foi realizado uma *survey*, em formato de questionário, que teve adesão de 33 programadores de diversos níveis de escolaridade e faixas etárias, que proveu embasamento para entender quais estratégias de ensino de programação, podem ser mais eficientes na visão dos programadores entrevistados. A partir das considerações

obtidas neste estudo, e, levando em conta o embasamento de que a colaboração é um fator relevante para o aprendizado, foi elaborada uma proposta de boas práticas para o ensino colaborativo de programação.

Este trabalho está estruturado da seguinte forma: na Seção 2 apresenta-se trabalhos que tem relação com este estudo. Na Seção 3 é exibido o estudo realizado para identificação do perfil de aprendizado dos estudantes de programação e os resultados obtidos neste estudo. Na Seção 4 apresenta-se a proposta de boas práticas para ensino colaborativo de programação e por fim, na Seção 5 apresentam-se as considerações finais e trabalhos futuros.

## 2. Trabalhos Relacionados

A ferramenta desenvolvida por Moura et al. 2017, denominada CoDiVision, para inferir o nível de contribuição de desenvolvedores em um projeto de software, pode servir de apoio aos professores para a realização de avaliações de estudantes em disciplinas ligadas ao ensino de programação. Esta ferramenta tem como objetivo principal a avaliação de estudantes em trabalhos desenvolvidos em grupo, durante uma disciplina visando ensinar o desenvolvimento de software.

Amaral et al. 2015 utilizaram a ferramenta Robocode no ensino de algoritmos e programação de computadores para alunos do ensino médio. Este estudo também envolveu alunos de graduação que já haviam cursado disciplinas de programação de computadores e que atuaram como tutores e coorientadores dos alunos do ensino médio. O ambiente Robocode foi utilizado como ferramenta de apoio ao ensino de algoritmos, baseando-se em uma estratégia lúdica de ensino, proporcionando um contato com a linguagem de programação Java ainda no ensino médio, com o objetivo de estimular o raciocínio lógico-matemático dos alunos de forma rápida e divertida. Os alunos obtiveram um aumento do interesse pela programação e desenvolveram a capacidade de liderança, proveniente da experiência em coorientar os alunos aprendizes.

O estudo realizado por Amaral et al. 2017 apresenta o projeto, a construção e a validação de uma ferramenta para o apoio ao processo de ensino e de aprendizagem de lógica e programação, para alunos iniciantes. Esta pesquisa foi baseada em conceitos e teorias de aprendizagem aplicadas em um Portal Web, por meio do qual o estudante pode realizar a construção do conhecimento sobre programação, de forma autônoma e à distância. O resultado destes experimentos foi o reconhecimento de um desempenho satisfatório dos alunos que utilizaram o ambiente para o aprendizado de lógica de programação.

A revisão sistemática de literatura desenvolvida por da Silva et al. 2014, tinha como objetivo investigar a eficácia do uso de jogos digitais como ferramenta para auxiliar o processo de ensino e aprendizagem de programação. Dos 29 artigos incluídos nesse estudo, 97% deles utilizavam-se de jogos como uma ferramenta eficaz no ensino e aprendizagem de programação. Este estudo concluiu que a utilização de jogos digitais ajudam no desempenho dos alunos, diminuindo as desistências e aumentando a motivação.

O estudo realizado por Marcolino and Barbosa 2015 teve o propósito de identificar os softwares e aplicações educacionais que contribuam no ensino de programação. O mapeamento sistemático de literatura produzido, permitiu o desenvolvimento de um catálogo de contribuições, que adotam diferentes estratégias de ensino na mitigação de problemas enfrentados nas disciplinas de programação. Os resultados deste mapeamento sistemático apresentaram as principais contribuições de softwares para promover o ensino

de programação e tópicos relacionados. Também foi possível identificar as principais lacunas a serem tratadas e inferir que há a falta de sinergia entre as diferentes iniciativas e instituições, na busca por soluções integradas, adaptáveis às exigências educacionais dos discentes e docentes e, com soluções que abranjam uma maior gama de problemas enfrentados. Embora os trabalhos relacionados descrevam importantes tópicos sobre ensino de programação, nenhum deles tem foco na percepção do aluno sobre o seu aprendizado, bem como nenhum apresenta uma proposta de boas práticas para os professores se basearem no ensino de programação.

### 3. Identificação do perfil de aprendizado dos estudantes de programação

Para auxiliar professores a guiar os seus trabalhos de preparação e execução de aulas, foi realizado um estudo exploratório visando identificar, sob o ponto de vista de programadores, qual a percepção destes em relação a forma como este conhecimento foi construído.

Foi desenvolvido um questionário, o qual tinha por objetivo identificar o perfil do respondente em relação à idade e escolaridade, com demais perguntas a respeito de como o respondente aprendeu a programar. Estas perguntas versaram sobre metodologias de ensino, formas de estudo coletivo ou individualizado, guiado ou autodidata. O questionário foi formado por questões objetivas e questões descritivas. O formato misto de questões permitiu o recebimento de respostas pontuais para perguntas nas quais buscava-se uma percepção exata acerca de um fenômeno, e ao mesmo tempo, obter uma percepção em amplo aspecto para questões cujas respostas podem ser diversas, dependendo da percepção do respondente.

É importante enfatizar que os respondentes concordaram de forma livre e esclarecida em participar do estudo e foi dada a estes a oportunidade de retirar a sua participação a qualquer momento, bem como garantido o anonimato das respostas. As perguntas efetuadas no estudo foram as seguintes:

1. Em relação à sua forma de aprendizado e relação com orientação de profissionais:
  - (a) Aprendo mais com ajuda de professor em sala de aula.
  - (b) Aprendo mais com ajuda de professor em videoaulas.
  - (c) Aprendo mais seguindo tutoriais criados ou disponibilizados pelo professor.
  - (d) Aprendo mais seguindo tutoriais em formato de videoaulas.
  - (e) Não aprendo com orientação de professor.
2. Analisando o seu aprendizado utilizando atividades práticas:
  - (a) Aprendo mais quando realizei atividades práticas em casa.
  - (b) Aprendo mais quando realizei atividades práticas individualmente em aula.
  - (c) Aprendo mais quando realizei atividades práticas com os colegas em aula.
  - (d) Aprendo mais realizando atividades práticas com auxílio do professor.
  - (e) As atividades práticas me confundem, aprendo mais quando não faço atividades práticas.
3. Em relação aos estudos realizados de forma individual, você acredita que:
  - (a) Aprende mais lendo tutoriais.
  - (b) Aprende mais assistindo videoaulas
  - (c) Aprende mais ouvindo (por exemplo, podcast).
  - (d) Aprendo mais fazendo atividades.

4. Em relação às metodologias ativas utilizadas no ensino de programação, com quais das seguintes metodologias, você teve contato (questão de múltipla escolha, onde os participantes poderiam selecionar mais de uma opção)?
  - (a) Programação em Pares (quando duas pessoas programam juntas).
  - (b) *Coding Dojo* (quando existe um piloto e um copiloto programando para resolver um desafio e os participantes ficam em silêncio, a cada rodada é trocado o piloto e o copiloto).
  - (c) Programação de projetos reais (por exemplo desenvolvimento de aplicações comerciais, sites, aplicativos que tem uma finalidade específica).
  - (d) Participação em feiras e eventos.
  - (e) Visitas técnicas.
  - (f) Estudos de caso (utilização de um exemplo real de algo que aconteceu na indústria para explicação em sala de aula).
  - (g) Utilização de analogias com a vida real para ensino de programação.
5. Sobre a metodologia ativa que você teve contato em aula, descreva sua experiência, fale sobre os pontos positivos e negativos.
6. Em relação a sua experiência com aprendizado de programação, o que você considera que poderia ser melhorado no processo de ensino para reduzir as suas dificuldades

Este estudo obteve 33 respostas, sendo estas 23 oriundas de programadores que possuem curso superior concluído ou em andamento; 8 programadores que possuem curso técnico concluído ou em andamento e um programador com ensino médio concluído ou em andamento.

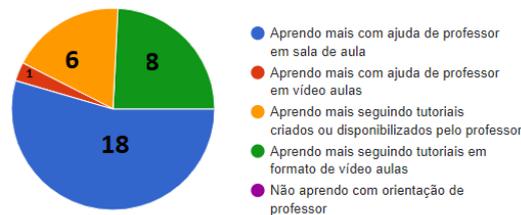
### 3.1. Análise dos dados

Em relação a aprendizagem com ajuda de profissionais, questionada na primeira pergunta, 18 dos 33 sujeitos participantes, responderam que aprendem mais com ajuda de professor em sala de aula, oito aprendem mais, seguindo tutoriais em formato de videoaula; Seis aprendem mais, seguindo tutoriais criados ou disponibilizados pelo professor e um aprende mais com ajuda de professor em videoaulas.

Analisando estas respostas, foi possível inferir que, a maior parte dos alunos aprende com interferência direta de profissionais. A maioria através do auxílio de professor em sala de aula, seguido por programadores que aprendem seguindo tutoriais disponibilizados em videoaulas, sendo que, um tutorial em formato de videoaula é parecido com uma aula presencial, com o diferencial que na videoaula, o aluno pode parar, reiniciar e assistir novamente quantas vezes desejar. A Figura 1 apresenta estas informações.

Em relação a realização de atividades práticas, a análise da segunda questão identificou que, a maioria dos programadores acredita que o aprendizado foi mais efetivo quando realizado através de atividades práticas em casa (14). Além disso, nove (9) programadores acreditam que aprenderam mais quando realizaram atividades práticas em aula com apoio de colegas, de forma colaborativa; Seis (6) programadores acreditam que aprenderam mais quando realizaram atividades práticas em aula individualmente e quatro (4) programadores acreditam que aprendem mais realizando atividades práticas com auxílio do professor.

Neste sentido, foi possível inferir que, todos os profissionais consideram a realização de atividades práticas importantes no processo de aprendizado de programação,



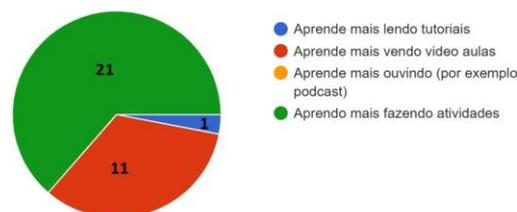
**Figura 1. Aprendizado com acompanhamento de profissional**

sendo que, o maior número de programadores respondeu que aprende mais realizando as atividades em casa. Esta resposta se contrapõe a anterior, onde foi identificada a importância da figura do professor no processo de aprendizagem. Contudo, acredita-se que as respostas tratam-se de dois momentos distintos: a introdução e apresentação dos conteúdos por parte do professor, e a realização das atividades em si. Realizando atividades em casa, o aluno tem mais tempo para desenvolver o seu raciocínio e menor possibilidade de interrupções. Além disso é importante perceber nestas respostas que 9 sujeitos enfatizaram que aprendem mais realizando atividades juntamente com os colegas, um fator que caracteriza o apoio do ensino colaborativo na aprendizagem. A Figura 2 mostra estas informações.



**Figura 2. Realização de Atividades Práticas**

Esta percepção é complementada pelas respostas obtidas a partir da terceira pergunta, que versou sobre, quando estão estudando individualmente, o que os programadores consideraram ser mais eficiente para o aprendizado, sendo que vinte e um (21) dos respondentes, consideram que aprendem mais fazendo atividades do que assistindo videoaulas ou lendo tutoriais. Estas informações são apresentadas na Figura 3.



**Figura 3. Prática x Leitura x Visualização**

Em relação as metodologias ativas utilizadas para ensino de programação, inferiu-se que os programadores respondentes tiveram oportunidade de interagir com diversas metodologias, destacando-se: programação em pares e desenvolvimento de projetos reais. A Figura 4 apresenta as respostas obtidas. É importante enfatizar que, esta pergunta foi de múltipla escolha, então os participantes podiam selecionar mais de uma alternativa.



**Figura 4. Práticas Utilizadas no Ensino de Programação**

Complementando estas respostas, obteve-se alguns *feedbacks* de forma descritiva na questão aberta onde era solicitado aos programadores, que avaliassem a sua experiência com as práticas. Nestas questões, despertou atenção a satisfação dos alunos com a realização de projetos reais e críticas a prática de *coding dojo*. Estas percepções são enfatizadas pelas seguintes frases escritas pelos participantes:

- “Um maior número de atividades voltadas a casos reais e aplicados a projetos existentes sempre facilitou meu aprendizado e me trouxe mais segurança para iniciar no mercado trabalho.”
- “Estudos de caso ou desafios foram as melhores experiências. Acredito que, por mais que a pessoa não tenha tanta facilidade em ‘aprender fazendo’, fazer uma atividade prática ajuda muito a fixar o que foi aprendido.”
- “Programação de projetos reais me trouxeram a visão de como as coisas realmente funcionam de como elaborar uma solução focada no problema.”
- “*Coding Dojo* com *Pair Programming* foi uma experiência ruim dada o desnível da turma. Não considero uma ideia válida para meu tipo de aprendizado.”
- “*Coding dojo* é muito bom para aprender, mas eu tenho problemas com outras pessoas me observando então acabo não tendo um desempenho tão bom.”

A segunda questão aberta, teve um cunho generalista, pedindo para os respondentes descreverem o que poderia ser melhorado no processo de ensino para reduzir as dificuldades na aprendizagem. Por ser uma questão aberta, as respostas foram bastante diversificadas, sendo as principais:

- Aumento da carga horária de aula.
- Mais prática de programação em aula.
- Mais exemplos para guiar a resolução de atividades.
- Redução do número de alunos nas turmas para que o professor possa acompanhar melhor, de forma individual, os alunos.
- Introdução de pequenos desafios para serem resolvidos pelos alunos.
- Aumentar o número de estudos de caso reais, ex: programação e banco de dados integrados em uma disciplina e não em disciplinas separadas.
- Caminhar prática e teoria juntas. Nunca separadas.
- Aumento do material de apoio: códigos comentados e sugestão de leituras.
- Evitar realização de aulas prática-expositivas, quando o professor digita o código para o aluno copiar.

Estas percepções podem ser exemplificadas pelas seguintes frases escritas pelos respondentes:

- “Mais exemplos e pequenos desafios. Maior número de desafios práticos, com casos reais, aos alunos, para desenvolver o raciocínio lógico e a fixação do conceito. Dificilmente se aprende apenas ouvindo e copiando.”

- “Maior número de desafios práticos, com cases reais, aos alunos, para desenvolver o raciocínio lógico e a fixação do conceito. Dificilmente se aprende apenas ouvindo e copiando.”
- “Mais prática em máquinas e menos programação em papel.”
- “Mais estudos de casos reais e resolução de problemas reais, mostrando a atividade por completo, ex: programação e banco de dados integrados em uma disciplina e não em disciplinas deparadas.”
- “Professor entender a curva de aprendizado do aluno, e principalmente não fazer o famoso ‘copiem o que estou fazendo na tela’.”

#### 4. Ensino Colaborativo de Programação

O ensino colaborativo possui diversas práticas e ferramentas que são utilizadas em sua aplicação. Esta Seção tem o objetivo de reforçar as práticas e ferramentas que o ensino colaborativo disponibiliza, com as respostas obtidas e descritas anteriormente, que refletem a visão dos programadores sobre como eles aprendem programação. Baseando-se neste critério, propõe-se o seguinte conjunto de boas práticas para ensino colaborativo de programação:

- Priorizar a realização de atividades práticas;
  - A realização de atividades práticas deve prioritariamente ser realizada em grupo. Para isso devem ser utilizadas ferramentas de versionamento e compartilhamento de código-fonte. Estas ferramentas permitem a programação em grupos, a distância de forma assíncrona e também permite um melhor controle do professor sobre a autoria de cada etapa do projeto. São exemplos de ferramentas que podem ser utilizadas para este fim, o SVN e Git.
- Disponibilizar aos alunos exemplos de soluções de problemas através de códigos-fonte comentados;
  - Para isso pode ser utilizado, além das ferramentas de versionamento e compartilhamento supracitadas, recursos como *blog*, fóruns, e *wikis*. Nestes três exemplos de recursos, os alunos poderão comentar os materiais disponibilizados e interagir entre si e com o professor.
- Disponibilizar aos alunos materiais de leitura sobre os conteúdos, priorizando materiais que possuam uma linguagem facilmente compreensível;
  - Da mesma forma que no item anterior, podem ser utilizados para este fim: *blog*, fórum e *wiki*. Também podem ser utilizados ambientes virtuais de aprendizagem, como por exemplo o Google Classroom. Esta ferramenta destaca-se porque permite que os alunos comentem os materiais disponibilizados pelos professores e disponibilizem materiais para os colegas.
- Utilizar analogias com o cotidiano do aluno no momento de explicar conteúdos; Neste sentido, o professor pode contar com a colaboração dos alunos de forma oral, incentivando os alunos a trazerem exemplos do seu dia a dia. Alternativamente, pode ser realizado com a turma a criação de um portfólio, onde cada aluno ou grupo de alunos possam contribuir com alguma vivência ou ponto de vista sobre um assunto. Os dados coletados no portfólio podem ajudar o processo de construção de exemplos. Um exemplo de ferramenta colaborativa que pode ser usado para construção de portfólio é o Padlet
- Utilizar videoaulas como material básico ou complementar;
  - As videoaulas podem ser disponibilizadas em ferramentas que permitam a interação dos alunos. *Blog* é um bom exemplo, mas também em ambiente virtual de aprendizagem que permita colaboração, que conforme supracitado, um exemplo é o Google Classroom.

- Utilizar tutoriais para atividades específicas;
  - Para disponibilização de tutoriais, novamente cita-se como exemplo de ferramentas colaborativas: *blog* e *wiki*, contudo, neste tópico é possível elencar como exemplo principal a *wiki* disponibilizada pelo *site* Github, que é um *site* que disponibiliza a criação de repositórios remotos de códigos-fonte que utilizam Git como ferramenta de versionamento. Ao utilizar a *wiki* desta ferramenta, os alunos podem ver o código-fonte e os tutoriais no mesmo local.
- Desafiar os alunos;
  - Para desafiar os alunos, diversas plataformas podem ser utilizadas. Entretanto, para que os desafios sejam colaborativos, é necessário utilizar ferramentas que permitam a criação de times para solução de desafios. Neste contexto, sugere-se o uso da ferramenta StudyPlay (Zanin et al. 2018), para criação de times e desafios.
- Trabalhar os projetos desenvolvidos em sala de aula baseando-se em casos de uso reais;
  - Aqui a colaboração novamente se dá através do trabalho em equipe e do compartilhamento de informações. Pode-se neste momento, pedir para que os alunos contribuam com casos de uso, novamente fazendo uso de portfólio, ou então através de uma apresentação de *slides* compartilhada em alguma plataforma de *cloud computing*.
- Evitar a realização de aula expositiva-dialogada e prática-expositiva (quando professor faz o código e o aluno copia).
  - As aulas com este formato não incentivam a colaboração, porque o aluno se torna um agente passivo. Pode-se alternar estas aulas com aulas que instiguem a colaboração e o protagonismo do aluno. São exemplos de técnicas que podem ser utilizadas: explorar a utilização de Jigsaw (quebra cabeças), tratando cada parte do código como um tópico a ser explorado por um grupo isolado (peça), e que juntos formam o todo; explorar a utilização de controvérsia acadêmica ou júri simulado, considerando diferentes versões de código para um mesmo problema e propondo debate, para decisão de qual melhor para resolver o problema.

## 5. Conclusões Finais e Trabalhos Futuros

Neste trabalho foi apresentado um estudo realizado para identificar os métodos que programadores acreditam terem sido os mais eficientes, entre os aplicados na sua formação. A partir da opinião dos programadores que participaram deste estudo, buscou-se ferramentas de trabalho colaborativo, que podem ser utilizadas para tornar o processo de ensino e aprendizado colaborativo de programação. A partir disso, foi elaborado um conjunto de boas práticas para o ensino colaborativo de programação. Acredita-se com isso que, tornando o ensino de programação colaborativo e utilizando estratégias efetivas de ensino, é possível melhorar o aprendizado, permitindo que os alunos se apropriem mais facilmente dos conteúdos.

Como continuação deste estudo, deverá ser realizado um experimento controlado, colocando em ação as práticas propostas e mensurando a melhoria no aprendizado dos alunos das turmas que seguirem este guia, comparado aos que não seguirem. Além disso, deverão ser efetuados estudos, visando aplicar as práticas propostas na ferramenta StudyPlay (Zanin et al. 2018), com o objetivo de torná-las facilmente aplicáveis.

## Referências

[Amaral et al. 2017] Amaral, E., Camargo, A., Gomes, M., Richa, C. H., and Becker, L.

(2017). Algo+ uma ferramenta para o apoio ao ensino de algoritmos e programação

V. 17 N° 3, dezembro, 2019 \_\_\_\_\_RENOTE

DOI:

- para alunos iniciantes. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 28, page 1677.
- [Amaral et al. 2015] Amaral, L., Braga, G., Pantaleão, E., et al. (2015). Plataforma robocódigo como ferramenta lúdica de ensino de programação de computadores-pesquisa e extensão universitária em escolas públicas de Minas Gerais. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 26, page 200.
- [Castro and Menezes 2011] Castro, A. and Menezes, C. (2011). *Aprendizagem Colaborativa com suporte computacional*, chapter 9, pages 135–153. Elsevier, Rio de Janeiro, 1 edition.
- [CBSI 2019] CBSI (2019). Computação é o que mais sofre evasão em universidades públicas e privadas. Acessado em: 16-06-2019.
- [da Silva et al. 2014] da Silva, T. R., Medeiros, T. J., and Aranha, E. H. d. S. (2014). Jogos digitais para ensino e aprendizagem de programação: uma revisão sistemática da literatura. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 25, page 692.
- [Ferreira and Reategui 2016] Ferreira, V. H. and Reategui, E. B. (2016). Uma comunidade de prática para o ensino e a aprendizagem de programação. *RENOTE: Revista Novas Tecnologias na Educação. Porto Alegre, RS. Vol. 14, n. 2 (dez. 2016), p.[1]-11.*
- [Giraffa and Mora 2013] Giraffa, M. M. and Mora, M. d. C. (2013). Evasão na disciplina de algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno. In *Conferencia Latino Americana Sobre E Abandono En La Education Superior.*
- [Gomes et al. 2008] Gomes, A., Henriques, J., and Mendes, A. (2008). Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. *Educação, Formação & Tecnologias-ISSN 1646-933X*, 1(1):93–103.
- [Infojobs 2019] Infojobs (2019). Vagas de emprego programador java. Acessado em: 07-07-2019.
- [Marcolino and Barbosa 2015] Marcolino, A. and Barbosa, E. F. (2015). Softwares educacionais para o ensino de programação: Um mapeamento sistemático. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 26, page 190.
- [Moura et al. 2017] Moura, V., Ibiapina, I., Lira, W. A. L., and Neto, P. S. (2017). Codivision: Uma ferramenta para apoio na avaliação de estudantes no ensino de programação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 28, page 1567.
- [Sparremberger et al. 2017] Sparremberger, A. S., Barbosa, J., and Gluz, J. (2017). Group-profile: Um modelo de gerenciamento de perfis de grupos de aprendizes. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 28, page 1687.
- [Wohlin 2014] Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, EASE '14*, pages 38:1–38:10, New York, NY, USA. ACM.
- [Zanin et al. 2018] Zanin, A., Sparremberger, A., Becker, T., and Barbosa, J. (2018). Studyplay: Um modelo gamificado para incentivo a realização de atividades extra-classe. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 29, page 1683.