

# Uso de avaliação por pares em disciplinas introdutórias de programação

Allan Lucio Correia<sup>1</sup>, Danilo Victor B. da Costa<sup>1</sup>, Alexandre de Andrade Barbosa<sup>1,3</sup>, Evandro de Barros Costa<sup>2,3</sup>

<sup>1</sup>Universidade Federal de Alagoas - Campus Arapiraca (UFAL)  
Caixa Postal 61 – Arapiraca - AL - Brasil

<sup>2</sup>Instituto de Computação - Universidade Federal de Alagoas (UFAL)  
Caixa Postal 15.064 - 91.501-970 - Maceió - AL - Brasil

<sup>3</sup>Dept. de Sistemas e Computação - Universidade Federal de Campina Grande (UFCG)  
Caixa Postal 10.106 – Campina Grande – PB – Brasil

{allan.correia,danilo.costa,alexandre.barbosa}@arapiraca.ufal.br,  
ebc.academico@gmail.com

**Abstract.** *Programming courses have a high rate of failures and dropouts. Several researchers report that the difficulty to achieve a fast and effective feedback, is one of the motivators of this problematic scenario. Fast feedback is important to allow learning of any concept. Some researchs propose the use of peer review as a means of providing feedback. In this paper is proposed a peer review approach used as part of programming courses. Currently a virtual learning environment is being adapted to include this proposal.*

**Resumo.** *As disciplinas de programação têm apresentado um alto índice de reprovações e desistências. Diversos pesquisadores descrevem que a dificuldade de obtenção de um feedback rápido e eficaz, é um dos motivadores deste cenário problemático. O feedback rápido, é importante para possibilitar o aprendizado de qualquer conceito. Algumas pesquisas propõem o uso de avaliação por pares (AP) como meio de prover feedback. Neste trabalho é proposta uma abordagem de avaliação por pares, como parte dos processos de ensino e aprendizagem de programação. Atualmente um ambiente virtual de aprendizagem está sendo adaptado para contemplar a proposta aqui descrita.*

## 1. Introdução

A programação é uma das competências fundamentais na área da computação. Contudo, apesar de sua importância, as disciplinas de programação têm apresentado um alto índice de reprovações e desistências [Cristovão 2008] [Yadin 2011]. Para o cenário apresentado em [Barbosa et al. 2014], o percentual de reprovações nas disciplinas de programação é de aproximadamente 50%, cenários similares são encontrados em outras instituições, tal como citado em [Barbosa et al. 2011] e [Noschang et al. 2014]. Por isso, atualmente existe uma crescente preocupação na comunidade acadêmica em relação a essas dificuldades.

Diversos pesquisadores descrevem que a dificuldade de um acompanhamento adequado por parte do professor para cada um dos alunos nas disciplinas de programação,

é um dos fatores motivadores dos diversos problemas existentes neste contexto [Moreira 2014] [Stegeman et al. 2014].

O *feedback* rápido é de extrema importância para possibilitar o aprendizado de qualquer conceito [Stegeman et al. 2014]. Com isso, pesquisas tem sido desenvolvidas com o intuito de propor métodos ou ferramentas para facilitar o acompanhamento das atividades dos discentes em disciplinas de programação. Algumas destas pesquisas, tais como [Raadt et al. 2007], [Sitthiworachart and Joy 2008] e [Moreira 2014], propõem o uso de avaliação por pares (AP) como meio de prover *feedback* rápido e eficaz. Este tipo de solução é amplamente utilizada em *Massive Open Online Courses (MOOCs)*, tal como descrito em [Kulkarni et al. 2013], onde os cursos são aplicados para milhares de pessoas, e assim como ocorre no contexto de programação, é custoso para o professor avaliar cada solução.

Neste trabalho é proposto o uso de uma abordagem de AP como parte integrante dos processos de ensino e aprendizagem de programação. Duas competências importantes para o aprendizado de programação são trabalhadas com o uso desta abordagem, sendo estas, a auto avaliação do código desenvolvido pelo aluno, comparando a sua solução com outras, e a análise crítica dos códigos criados por outros discentes.

Este artigo está estruturado da seguinte maneira, na sessão 2 são descritos os conceitos relacionados com a avaliação em pares, na sessão seguinte os trabalhos correlatos são apresentados, na sessão 4 é exibida a proposta de avaliação em pares para o contexto das disciplinas introdutórias de programação, finalmente na última sessão, são exibidas as conclusões e os trabalhos futuros.

## **2. Avaliação por pares**

Segundo [Rust et al. 2003] inúmeras pesquisas comprovam que, alguns estudantes não conseguem obter o rendimento esperado nas disciplinas, devido a sua baixa compreensão acerca dos conteúdos exigidos. Por outro lado, de acordo com [Stipek 1993], existem alunos que entendem os conteúdos rapidamente e precisam de novos desafios, ou acabam ficando desmotivados. Visto isso, facilitar as discussões e a troca de informações entre os alunos é uma maneira bastante eficaz para maximizar o aprendizado.

A avaliação por pares, em inglês *peer assessment* ou *peer review*, é um método avaliativo, onde os alunos tem responsabilidades que tradicionalmente pertencem apenas aos professores. Nessa metodologia cada aluno será receptor e transmissor de conhecimentos. A metodologia coloca o aluno em situações de confronto, uma vez que é preciso refletir sobre sua solução em relação as outras soluções observadas, o que estimula o processo de aprendizado, pois exige do aluno as capacidades de argumentação, discussão, recepção de informações, exposição de ideias [Santos 2002] [Topping 1998] [Van Lehn et al. 1995]. Diferentemente do que ocorre na abordagem de ensino tradicional, onde o aluno é apenas receptor de informações.

Muitas vezes a AP pode ser confundida com outra abordagem de ensino aprendizagem, realizada através da interação entre os pares, conhecida como *Peer Instruction (PI)*. As aulas que seguem esse modelo geralmente são curtas e o professor foca em fazer os alunos desenvolverem o seu próprio conceito acerca do que está sendo estudado, e posteriormente incentivá-los a discutirem uns com os outros afim de chegarem a uma solução, e por fim apresentá-la ao professor [Crouch and Mazur 2001].

Na abordagem tradicional, o professor, mesmo quando assessorado por monitores, não consegue fornecer um *feedback* adequado e rápido para cada uma das soluções propostas pelos alunos, pois a correção de cada solução demanda uma quantidade de tempo considerável. Sendo assim, o uso de AP pode reduzir a carga de trabalho sob o professor, trazendo *feedback* mais rápido aos alunos e fazendo com que o professor possa se concentrar em outras atividades pedagógicas [Raadt et al. 2007][Kulkarni et al. 2013].

Porém os benefícios relacionados com a aplicação do método de AP dependem fortemente de duas características, a seleção dos avaliadores e a qualidade dos comentários fornecidos pelos pares. A seleção dos avaliadores para os trabalhos submetidos é importante, pois, segundo [Fry 1990], caso a seleção seja inadequada, a efetividade será comprometida, o que poderá prejudicar alguns alunos durante o processo. A qualidade dos comentários fornecidos pelos pares é importante pois, caso o comentário seja pouco esclarecedor, nenhum tipo de novo conhecimento será obtido pelo avaliado.

Em [Kulkarni et al. 2013], é citado que, em geral, existe alta correlação entre as avaliações fornecidas por alunos e professores/tutores, porém muitas vezes nas avaliações dos alunos são fornecidas notas 7% superiores à avaliação dos professores/tutores.

Desta maneira, pode ser descrito que a abordagem de avaliação por pares é promissora, existem contudo cuidados que devem ser tomados em relação a alocação dos avaliadores aos trabalhos, além de melhorias e correções que podem ser aplicadas para a obtenção de discussões mais ricas e avaliações mais acuradas.

### 3. Trabalhos correlatos

Nesta sessão, são exibidos os trabalhos encontrados na literatura que tratam sobre o uso de AP em contextos de ensino de programação. Visto isso, serão apresentadas descrições para os trabalhos de Raadt, Lai e Watson [Raadt et al. 2007], Sitthiworachart e Joy [Sitthiworachart and Joy 2008] e Moreira [Moreira 2014].

Outros trabalhos foram encontrados na literatura, com aspectos similares a proposta foco deste artigo. Contudo, estes não serão apresentados nesta sessão, pois diferem do contexto de aplicação ou não possuem como foco da nossa proposta. No trabalho de [Warren et al. 2014], é utilizada uma abordagem de avaliação por pares em contexto de programação, porém a abordagem de AP não é o foco do artigo. Chinn, em [Chinn 2005], apresenta uma abordagem de AP, para um contexto de ensino de análise de algoritmos, e não sobre o ensino introdutório de programação. O trabalho de [Sirotheau et al. 2011], propõe um ambiente de AP no ensino de programação, porém seu foco é a descrição de um avaliador automático de código. O trabalho [Kulkarni et al. 2013], está relacionados ao uso de uma abordagem de AP dentro de um curso de Interação Humano Computador ofertado em um *MOOC*, sendo portanto aplicado a um contexto diferente desta proposta.

Em [Raadt et al. 2007], é apresentada a ferramenta PRAISE, que a partir de 2006 foi utilizada no contexto de programação. Cada avaliação conduzida no PRAISE é realizada seguindo um conjunto de critérios estabelecidos pelo professor, ao final da verificação dos critérios o aluno deve fornecer um comentário. A partir de então os alunos podem submeter um documento com a solução, realizar duas avaliações e aguardar o *feedback*. No sistema proposto, as atividades ficam disponíveis do início até a data limite da atividade. Durante e após o período o professor pode acompanhar e moderar todas as

ações dos alunos. O sistema permite que o aluno avalie e submeta ao mesmo tempo, com exceção dos quatro ou cinco primeiros alunos a efetuar as submissões. Neste trabalho são identificados conflitos entre as avaliações, quando algum aluno não concorda com a avaliação recebida, o professor deve realizar moderação. Após o final do processo um *e-mail* é enviado ao aluno, este pode então observar o comentário e a avaliação recebida.

Em [Sitthiworachart and Joy 2008], O processo de AP descrito possui quatro etapas, realizadas de forma sequencial. Na primeira etapa, os alunos submetem uma solução, esta será avaliada automaticamente por testes unitários. A seguir, o trabalho submetido pelo discente será avaliado por três pares, os mesmos irão avaliar a qualidade do programa. Na terceira etapa, a qualidade de cada avaliação deve ser observada, para isso é necessário responder um conjunto com perguntas sobre diferentes critérios. Nestas duas etapas anteriores, os alunos devem discutir anonimamente e em grupo. Finalmente, os alunos podem observar sua nota e o *feedback* fornecido pelos pares. Qualquer observação fornecida em uma avaliação pode ser revisada em qualquer momento anterior ao *deadline* da atividade. Para identificar possíveis problemas nas avaliações um desvio padrão é adotado, caso uma avaliação tenha variação superior ao desvio padrão, o professor será acionado para realizar uma nova avaliação.

Na proposta contida em [Moreira 2014], descreve um sistema de AP, que foi utilizado por 3 semestres letivos. Esse sistema contempla as seguintes funcionalidades: criação de turmas, questionários e questões. Um questionário contém um conjunto de questões, que os alunos deverão responder. Cada questionário possui um intervalo de datas limite, fornecidos pelo professor. A ferramenta possui um identificador de plágio simples, que conta com intervenção do professor. Como padrão, uma solução é avaliada anonimamente por dois outros alunos, opcionalmente as soluções podem ser avaliadas por um monitor. Cada avaliação possui três opções: correto; parcialmente correto; e incorreto. O avaliador deve fornecer um comentário apenas se julgar a solução como parcialmente correta ou incorreta. Caso ocorram impasses entre as avaliações o professor deverá avaliar a questão, sendo sua avaliação a que possui maior peso.

#### **4. Proposta de utilização de avaliação por pares em disciplinas introdutórias de programação**

A interação entre humanos é descrita como essencial para o aprendizado em muitos domínios, incluindo o aprendizado introdutório de programação [Warren et al. 2014]. Para os cursos de programação presenciais o contato entre os estudantes ocorre no dia a dia, possibilitando, por exemplo, o debate dos problemas apresentados em listas de exercícios, nas soluções criadas e na formação de grupos para realização de trabalhos. O uso de AP pode enriquecer a interação entre os envolvidos.

Para estimular a assimilação dos conteúdos, a utilização de listas de exercícios é bastante comum em disciplinas de programação, pois a prática de programação é crucial para o aprendizado. Muitos pesquisadores defendem também que o aprendizado de programação passa pela leitura e compreensão de códigos de terceiros. Através de uma abordagem de AP ambas características podem ser obtidas. O professor pode se dedicar a elaboração de novas questões, ou a escolha de problemas propostos por terceiros, enquanto os alunos terão que observar, compreender e avaliar os códigos de seus pares.

Nesta sessão será apresentada a proposta de utilização de avaliação por pares em

disciplinas introdutórias de programação. A apresentação estará dividida em cinco seções, cujos conteúdos são: a descrição de limitações e dificuldades existentes nos trabalhos correlatos; os processos propostos na abordagem de AP descrita nesse trabalho; as estratégias para seleção de avaliadores para as soluções submetidas; os mecanismos de incentivo e regulação disponíveis; e finalmente as limitações e dificuldades associadas a esta proposta.

#### **4.1. Limitações e dificuldades existentes nos trabalhos correlatos**

Todos os trabalhos analisados descrevem uma abordagem de AP com processos fixos, não é possível adicionar um processo ou alterar a forma como estes são realizados. Os trabalhos de [Raadt et al. 2007], [Sitthiworachart and Joy 2008] e [Moreira 2014] são aplicações *web* cujo código não é disponibilizado para download. Portanto não podem ser modificados para atender uma realidade diferente do contexto, para os qual foram criados.

A possibilidade de discussão é um dos fatores que mais gera aprendizado no contexto da AP. O trabalho de [Sitthiworachart and Joy 2008], é o único, dentre os trabalhos correlatos encontrados, que descreve um processo onde um grupo de alunos pode discutir as soluções e as avaliações destas. Em [Raadt et al. 2007] e [Moreira 2014], um avaliador pode fornecer um comentário sobre o código avaliado, mas nenhum tipo de resposta ao comentário pode ser realizada. Desta forma, não ocorre discussão entre avaliador e avaliado, minimizando o potencial de aprendizado do aluno com seus pares.

A seleção dos avaliadores para as soluções submetidas é uma das características mais importantes dentro de uma abordagem de AP. Quando a seleção dos avaliadores não é realizada de forma adequada, o aprendizado pode ser comprometido, pois, por exemplo, um aluno com baixa compreensão do assunto pode ser alocado para avaliar o trabalho de um outro discente na mesma situação. No trabalho de [Raadt et al. 2007], é permitido que o aluno submeta e avalie as soluções em um mesmo momento, sendo que a alocação de avaliadores obedece a uma fila de submissões. Em [Sitthiworachart and Joy 2008], existem discussões em grupos, porém não é citado como estes grupos são formados. No trabalho de [Moreira 2014], o avaliador deve observar duas soluções e fornecer sua avaliação, não é citado no artigo como ocorre esta seleção. Desta forma, é possível concluir que em nenhum dos trabalhos correlatos analisados existe uma preocupação em selecionar os avaliadores adequados para cada trabalho.

A qualidade dos comentários gerados sobre as soluções é determinante, pois com isso um aluno com dificuldade, pode compreender algum aspecto que não havia entendido durante as aulas. No trabalho de Moreira [Moreira 2014], a avaliação do avaliador ocorre de forma automática, e está restrita a adequação da avaliação final, além disso os comentários fornecidos ao longo dos processos não são considerados. Em [Raadt et al. 2007], um aluno pode indicar se concorda ou não com a avaliação final obtida, com isso o professor deve resolver esse impasse. O trabalho de [Sitthiworachart and Joy 2008], é o único em que a avaliação dos comentários é efetuada, ocorrendo através de discussões anônimas em grupos.

Nos trabalhos observados, não existem mecanismos para regular ou incentivar ações, tal como fornecer um comentário no tempo adequado. Apenas em [Raadt et al. 2007], existe a funcionalidade de envio automático de *e-mails* informativos.

## **4.2. Protocolos e processos da abordagem proposta**

A abordagem de AP pode conter diversas atividades diferentes em sua aplicação, para cada atividade uma forma de execução pode ser determinada. Neste contexto, descrevemos um protocolo como a ordenação de um conjunto de processos, um processo deve ser compreendido como uma atividade e as restrições a ela aplicáveis.

Uma vez que um dos objetivos deste trabalho é propor uma abordagem de AP que possa ser adaptável a diferentes contextos, é descrito aqui, o conjunto de processos que poderá ser utilizado para formar o protocolo de aplicação desejado pelo professor. Desta forma, a especificação do protocolo deverá ocorrer através da inclusão dos processos desejados, da configuração de cada processo e finalmente da especificação das datas limite associadas a cada um destes.

Anteriormente a criação de qualquer atividade é necessário criar uma disciplina informando seus dados (ex. nome e carga horária) e aprovar à participação de todos os alunos matriculados. Após esse cadastro o professor deverá indicar as atividades, especificar as configurações desejadas e informar as datas limite, determinando assim um processo que limita os protocolos.

### **4.2.1. Criação de exercícios**

Para esta atividade o professor deverá postar um documento contendo a descrição do exercício, bem como especificar cada questão. Para cada questão devem ser fornecidos os critérios de avaliação e o peso de cada um na composição da nota. O professor deve também informar qual estratégia de seleção dos avaliadores será adotada, alocação automática ou manual. Caso deseje, o professor pode especificar os mecanismos de regulação e incentivo associados a atividade. Desta forma, por exemplo, será possível especificar uma penalidade, por não cumprir tarefas. Ao finalizar o professor deve determinar o período de início e fim da atividade, limitando o período associado aos processos posteriores.

### **4.2.2. Submissão**

Para esta atividade o aluno deverá submeter no sistema um arquivo, ou um conjunto de arquivos, que correspondem a solução proposta para o exercício. O professor pode restringir algumas características dos arquivos (ex. extensão e tamanho máximo). O professor pode configurar para que, caso o aluno não realize esta atividade, ficará impedido de prosseguir. Estes arquivos estarão disponíveis, para que os pares os avaliem num processo posterior.

### **4.2.3. Seleção dos avaliadores**

Para esta atividade, caso o professor tenha escolhido uma forma de alocação manual, este deve alocar os trabalhos que serão atribuídos, aos avaliadores. Caso tenha escolhido uma alocação automática, o próprio sistema irá selecionar os avaliadores, adotando para isso um algoritmo de recomendação de pares, que será descrito na seção 4.3, ou adotando uma alocação aleatória, se o professor desejar.

#### **4.2.4. Discussão entre pares**

Para esta atividade o professor poderá adotar uma forma de aplicação de discussão sem adoção de mecanismos de incentivo e regulação, ou seja, apenas as datas limite deverão ser fornecidas, ou com a adoção de mecanismos de incentivo e regulação, neste caso configurações adicionais serão necessárias.

#### **4.2.5. Grupos de discussão**

Os grupos de discussão, tem como foco estimular a troca de informações entre os alunos envolvidos na atividade, e assim como ocorre no processo de discussão entre pares, podem ser especificados os mecanismos de incentivo e regulação que serão utilizados. Para esta atividade o professor deverá indicar a quantidade máxima de integrantes de um grupo, especificar a estratégia de alocação de membros do grupo, e cadastrar o período em que esse grupo ficará aberto, para que os estudantes discutam com os demais membros.

#### **4.2.6. Nota**

Para esta atividade o aluno deverá fornecer uma nota para todos que avaliar. Cada nota será formada, pela avaliação relacionada aos critérios definidos pelo professor. A nota final do exercício será formada pela composição das notas de todos os avaliadores. O professor pode especificar um cálculo para obtenção da nota, descrevendo, se por exemplo, será utilizada a média ou uma mediana das notas dos avaliadores.

#### **4.2.7. Avaliação do avaliador**

Para esta atividade o estudante avaliado deverá dar uma nota a cada um dos pares que o avaliaram. Através da avaliação do avaliador será possível definir uma reputação para o aluno, como avaliador. Tomando como base essa reputação, é possível criar uma métrica para seleção de pares. Além disso, conflitos poderão ser identificados, por exemplo, quando comentário e nota forem inconsistentes.

### **4.3. Seleção de avaliadores**

Conforme descrições anteriores, acreditamos que a seleção dos avaliadores para as soluções submetidas é uma das características mais importantes dentro de uma abordagem de AP. Uma seleção de avaliadores adequada deve misturar alunos com perfis diferenciados e diferentes níveis de entendimento do conteúdo trabalhado. Desta forma, um aluno com dificuldades, pode esclarecer suas dúvidas, enquanto alunos com boa compreensão do conteúdo deverão aprofundar seu conhecimento, para fornecer uma boa argumentação.

Para identificar qual a melhor estratégia de pareamento, é necessário descobrir o nível de entendimento do aluno sobre o conteúdo. Para realizar esse processo serão coletadas métricas sobre o discente, em relação a atividades anteriores realizadas na abordagem de AP, bem como dados de suas sessões de estudo externas ao sistema. Para coleta de dados das sessões de estudo, externas ao sistema, uma abordagem similar a adotada em [Araujo et al. 2013] será utilizada.

#### **4.4. Mecanismos de incentivo e regulação**

Nos trabalhos correlatos, o resultado final obtido nestas avaliações, é utilizado como parte da média da disciplina relacionada. Além do peso na média final do discente, o único mecanismo utilizado para motivar os alunos a efetuarem as avaliações das soluções, descrito nestas iniciativas, é o envio de e-mails de aviso.

Acreditamos que alguns mecanismos de incentivo e regulação, podem ser aplicados na abordagem de AP, quando ela é aplicada ao contexto de programação. Tais mecanismos podem ser utilizados não somente, para lembrar ou forçar o aluno a fornecer os comentários em uma avaliação, mas em diversos outros processos presentes na abordagem.

As pendências de avaliação podem ser notificadas por *e-mail*, tal como mostrado nos trabalhos correlatos, e em uma área de notificações no próprio sistemas de AP. A área de notificações pode conter diversas informações, tais como data limite para submissão de códigos, informações sobre postagens do professor, indicação de comentário sobre uma solução proposta pelo aluno, entre outras. A área de notificações será exibida na tela inicial do usuário e pode ser visualizada a qualquer momento na aplicação.

Em [Moreira 2014], é exposto que frequentemente os alunos perdem prazos. Visto isso, para cada processo estabelecido pelo professor, ele poderá indicar se existirão penalidades, aos alunos que não cumprirem as metas, se sim, inserir um percentual de decréscimo na nota. No caso de comentários, muitas vezes os alunos argumentam muito próximo a data limite, é proposto que a discussão possa ocorrer em fases alternadas de comentário e replica. Assim, cada envolvido terá seu próprio tempo para argumentar.

Para qualquer processo que contenha como uma de suas ações a especificação de comentários, será possível para o aluno indicar a qualidade do comentário, bem como, solicitar a moderação do professor, notificando que algo está errado.

A qualidade da avaliação fornecida pelos discentes pode ainda ser verificada de maneira automática, através do uso de testes de unidade, tal como proposto em [Sitthiworachart and Joy 2008]. Contudo, na proposta contida neste artigo, o professor poderá indicar em que momento e de qual forma os testes devem ser utilizados, por exemplo, o processo de correção automática pode ocorrer antes da fase de submissão, após a avaliação fornecida por um avaliador, ou em qualquer outro momento.

Outros mecanismos de incentivo e regulação podem ser pesquisados. Diversos trabalhos tem como proposta, o uso dos conceitos de Gamificação em ambientes similares.

#### **4.5. Limitações e dificuldades**

Na abordagem proposta o professor deverá cadastrar os exercícios que serão avaliados através da AP, desta forma pode existir uma limitação em relação a quantidade de problemas disponíveis aos alunos. Tal limitação pode ser solucionada através da integração da abordagem proposta com um banco de problemas de programação, tal como em [Paes et al. 2013]. Contudo, essa integração pode restringir as adaptações desejadas para os diferentes contextos das instituições, o sistema de AP não funcionará integrado a um banco de questões.

Em [Raadt et al. 2007], os autores descrevem que caso a AP seja utilizada de forma inadequada, a retenção de estudantes pode aumentar. A solução para esta difi-



culdade é proposta em [Warren et al. 2014], neste trabalho os autores utilizam dois tipos de atividades que requerem avaliação, os quizzes e os mini projetos. Dentre as atividades citadas apenas os mini projetos são avaliados através de AP. Desta forma, os alunos não serão sobrecarregados e a abordagem pode ser utilizada de forma adequada.

Um outro problema que pode ocorrer com a aplicação de AP no contexto proposto, é a possibilidade de aumento na ocorrência de plágio das soluções. Contudo, [Moreira 2014], propôs o uso de identificadores de plágio simples, demonstrando que esta é uma solução eficaz para o contexto. Outras soluções amplamente utilizadas, tais como MOSS [Bowyer and Hall 1999] e GPLAG [Liu et al. 2006], poderão ser integradas ao sistema. Contudo, esta não é uma das funcionalidades propostas originalmente.

## 5. Conclusões e trabalhos futuros

No presente artigo foi apresentada uma proposta para utilização de uma abordagem de avaliação por pares como parte dos processos de ensino e aprendizagem de programação. Acreditamos que o uso desta abordagem possibilitará aos estudantes um *feedback* mais ágil sobre suas soluções, gerando assim um melhor aprendizado.

Uma vez que o evento se constitui como um importante fórum de debates acerca do ensino e aprendizagem de Computação, apresentamos este ensaio como uma proposta para a reflexão e o debate entre os professores, estudantes e demais autores que se relacionem com processos similares.

A pesquisa associada a este artigo ainda está em estágio inicial. Desta forma, a efetividade da proposta ainda será avaliada através de experimentos controlados. Estes experimentos serão executados com o apoio de um ambiente virtual de aprendizagem (AVA), que atualmente esta sendo adaptado para contemplar a abordagem descrita. O AVA citado possui código livre, pois a intenção dos pesquisadores envolvidos é que após a obtenção dos resultados iniciais a ferramenta seja disponibilizada para comunidade.

## Referências

- Araujo, E. C. d., Gaudencio, M., Menezes, A., Ferreira, I., Ribeiro, I., Fagner, A., Ponciano, L., Morais, F., Guerrero, D. S., and Figueiredo, J. A. (2013). O papel do hábito de estudo no desempenho do aluno de programação. In *Anais do CSBC/XXI WEI*.
- Barbosa, A. d. A., Ferreira, D. Í., and Costa, E. B. (2014). Influência da linguagem no ensino introdutório de programação. In *Anais do CBIE/25º SBIE*, pages 612–621.
- Barbosa, L. S., Fernandes, T. C., and Campos, A. M. (2011). Takkou: Uma ferramenta proposta ao ensino de algoritmos. In *Anais do CSBC/XIX WEI*.
- Bowyer, K. and Hall, L. (1999). Experience using "moss" to detect cheating on programming assignments. In *FIE '99. 29th Annual*, volume 3, pages 13B3/18–13B3/22 vol.3.
- Chinn, D. (2005). Peer assessment in the algorithms course. *SIGCSE*, 37(3):69–73.
- Cristovão, H. M. (2008). Aprendizagem de algoritmos num contexto significativo e motivador: um relato de experiência. In *Anais do CSBC/XVII WEI*.
- Crouch, C. H. and Mazur, E. (2001). Peer instruction: Ten years of experience and results. *American Journal of Physics*, 69(9):970–977.

- Fry, S. A. (1990). Implementation and evaluation of peer marking in higher education. *Assessment and evaluation in higher education*, 15(3):177–189.
- Kulkarni, C., Wei, K. P., Le, H., Chia, D., Papadopoulos, K., Cheng, J., Koller, D., and Klemmer, S. R. (2013). Peer and self assessment in massive online classes. *ACM Trans. Comput.-Hum. Interact.*, 20(6):33:1–33:31.
- Liu, C., Chen, C., Han, J., and Yu, P. S. (2006). Gplag: Detection of software plagiarism by program dependence graph analysis. In *Proceedings of the 12th ACM SIGKDD, KDD '06*, pages 872–881, New York, NY, USA. ACM.
- Moreira, B. G. (2014). Desenvolvimento de uma ferramenta de avaliação por pares para disciplinas de algoritmos e programação. In *Anais do CSBC/XXII WEI*.
- Noschang, L. F., Fillipi Pelz, E. A., and Raabe, A. L. (2014). Portugol studio: Uma ideia para iniciantes em programação. In *Anais do CSBC/XXII WEI*, pages 535–545.
- Paes, R. B., Malaquias, R., Guimaraes, M., and Almeida, H. (2013). Ferramenta para a avaliação de aprendizado de alunos em programação de computadores. In *Anais do 2º CBIE/WCBIE*, Campinas, SP.
- Raadt, M., Lai, D., and Watson, R. (2007). An evaluation of electronic individual peer assessment in an introductory programming course. In *Proceedings of the Seventh Baltic Sea Conference on Computing Education Research - Volume 88*, Koli Calling '07, pages 53–64, Darlinghurst, Australia, Australia.
- Rust, C., Price, M., and O'DONOVAN, B. (2003). Improving students' learning by developing their understanding of assessment criteria and processes. *Assessment & Evaluation in Higher Education*, 28(2):147–164.
- Santos, L. (2002). Auto-avaliação regulada: porquê, o quê e como?
- Sirotheau, S., Brito, S., Silva, A., Eliasquevici, M. K., Favero, E. L., and Tavares, O. (2011). Aprendizagem de iniciantes em algoritmos e programação: foco nas competências de autoavaliação. *Anais do 22º SBIE*.
- Sitthiworachart, J. and Joy, M. (2008). Computer support of effective peer assessment in an undergraduate programming class. *JCAL*, 24(3):217–231.
- Stegeman, M., Barendsen, E., and Smetsers, S. (2014). Towards an empirically validated model for assessment of code quality. In *Proceedings of the 14th Koli Calling International Conference on Computing Education Research*, pages 99–108. ACM.
- Stipek, D. J. (1993). Motivation to learn: From theory to practice.
- Topping, K. (1998). Peer assessment between students in colleges and universities. *Review of educational Research*, 68(3):249–276.
- Van Lehn, K., Chi, M., Baggett, W., and Murray, R. (1995). Progress report: Towards a theory of learning during tutoring. *Pittsburgh, PA: LRDC, University of Pittsburgh*.
- Warren, J., Rixner, S., Greiner, J., and Wong, S. (2014). Facilitating human interaction in an online programming course. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education, SIGCSE '14*, pages 665–670, NY, USA. ACM.
- Yadin, A. (2011). Reducing the dropout rate in an introductory programming course. *Acm Inroads*, 2(4):71–76.