# Desenvolvimento de uma ferramenta de avaliação por pares para disciplinas de algoritmos e programação

# Benjamin Grando Moreira<sup>1</sup>

<sup>1</sup>Centro de Engenharias da Mobilidade – Universidade Federal de Santa Catarina (UFSC) – Joinville – SC – Brasil

Benjamin.grando@ufsc.br

Resumo. A aprendizagem de algoritmos e programação apresenta desafios para o ensino e é assunto recorrente entre professores das disciplinas relacionadas, assim como a dificuldade na realização das avaliações. Esse artigo apresenta a estratégia de avaliação por pares e sua aplicação através de uma ferramenta em disciplinas de algoritmos e programação. O artigo descreve a ferramenta desenvolvida que é responsável por controlar as respostas dos alunos e sua distribuição para serem avaliadas. Também são apresentas discussões sobre alterações na ferramenta decorrentes de sua utilização e melhorias necessárias para atender melhor as necessidades.

#### 1. Cenário de Uso

Segundo [Barcelos, Tarouco e Berch 2009], o estudo de algoritmos é uma área fundamental no campo da computação e das engenharias. A aprendizagem de algoritmos constitui um dos problemas enfrentados pelos alunos que ingressam nestes cursos, sendo a disciplina que aborda tais conhecimentos uma das que tem maior índice de reprovação.

Disciplinas como algoritmo e programação de computadores exigem dos alunos raciocínio lógico, resolução de problemas e capacidade de abstração da solução em uma representação formal e/ou em uma linguagem computacional [Jesus e Brito 2010].

Por seu elevado índice de reprovação e desistência em instituições de ensino no Brasil, a disciplina de algoritmos tem sido foco dos professores, preocupados com a melhoria no processo de ensino e a qualidade oferecida para os alunos, mostrando assim necessidade de algumas alterações tanto na parte didática como na metodológica da apresentação [Rodrigues Junior 2004]. Sendo assim, é primordial que sejam desenvolvidos aplicativos e metodologias que contribuam para a melhoria do processo de ensino-aprendizagem destes alunos [Jesus e Brito 2010].

Segundo [Monteiro e Fragoso 2005], a avaliação entre pares é um instrumento de avaliação educacional e permite alcançar objetivos pedagógicos diversos, fazendo parte de uma avaliação formadora em que aqueles que aprendem são construtores do seu próprio conhecimento.

A avaliação por pares enfatiza a aprendizagem colaborativa, ao permitir que os estudantes se avaliem mutuamente e, ao ter no retorno uma das suas dimensões estruturantes, contribui para dinamizar o diálogo entre o professor e os estudantes e

estimula o debate entre eles, além de melhorar suas capacidades críticas e argumentativas [Alves 2012].

Segundo [Topping 1998], para estudantes do ensino superior de mesmo período e formação, a avaliação por pares é um arranjo onde os estudantes consideram a quantidade, o nível, o valor, a qualidade ou o sucesso dos resultados de aprendizagem de seus pares.

#### 2. Desenvolvimento

A ferramenta que foi desenvolvida é uma ferramenta de internet com acesso via navegador e que permite o gerenciamento de turmas e criação de questionários. A linguagem de programação utilizada foi a PHP com banco de dados MySQL. O desenvolvimento da parte visual do sistema utilizou o framework Bootstrap.

O software foi utilizado por 3 semestres letivos. A versão atual é uma versão melhorada da aplicada pela primeira vez e uma nova versão está em desenvolvimento, com a reescrita da aplicação e implementação de melhorias identificadas nas duas aplicações seguintes. Nos 3 semestres de aplicação, a nota obtida no software compôs parte da média final do aluno na disciplina.

## 3. Apresentação do software

atualmente acessada ferramenta através do endereço http://galirows.com.br/a3a/. O sistema será disponibilizado como um servico, sendo preciso criar uma conta, e não sendo disponibilizado para download e instalação. O acessado demonstrativo ser através pode http://galirows.com.br/a3a/videoMSIE2014.zip. apresentadas Α seguir são funcionalidades da ferramenta.

A ferramenta permite o gerenciamento de turmas e criação de questionários. O professor também pode cadastrar questões disponibilizando-as de forma pública ou privada, além de poder associar gabarito em diversas linguagens de programação. Também é possível associar um link que pode complementar a solução algorítmica, como por exemplo, um vídeo explicativo.

Questões publicadas como públicas estarão disponíveis para todos os professores utilizadores do sistema, bem como demais elementos vinculados à questão (gabaritos e links). Professores que adicionarem novos gabaritos a uma questão pública terão também o gabarito disponibilizado para todos os professores. Essa não foi uma implementação intencional, mas foi mantida, permitindo uma forma colaborativa para compartilhamento de algoritmos.

Além de questões o professor deve cadastrar os questionários que serão aplicados para cada turma. Um questionário é composto de uma ou mais questões previamente cadastradas e possui datas limites de início e finalização definidas pelo professor. O professor também deve definir a linguagem de programação que será utilizada no questionário, sendo a definição da linguagem utilizada para a geração do recurso de *syntax highlight* (recurso que destaca com cores distintas as palavras que são próprias da uma linguagem de programação).

A Figura 1 mostra uma visão geral do sistema, que possui o menu de opções a esquerda e sua área de conteúdo a direita (nas demais telas demostradas no artigo o menu estará suprimido). Nessa tela são mostradas as turmas do professor com um ícone de uma mão apontando para baixo, que ao se clicada mostra efetivamente os questionários disponibilizados para a turma. De cada questão destaca-se a porcentagem de avaliações respondidas pelos alunos e sua porcentagem de avaliação por parte dos próprios aluno.

MENU		Listar questionário	os				
Início			D	emover 🗶 l	Editar: 🕼 I Adicionar quest	ões: ☀   Responder: ≣   Ver	Legenda:
Questões	Þ	ABCD (2013/2) &	10	ciliovei. 🕶 į	Editar. 6   Adicional quest	ocs. 🐷   Nesponder. 🔤   Ver	resposias. •   Flagios. •
Questionários	<b>&gt;</b>	EFG (2013/2) 👨					
Ver notas		Título	Feitos	Avaliados	Data de aplicação	Data de encerramento	Ação
Editar dados		Questionário 7	42%	1%	2013-12-13 00:01:00	2013-12-16 23:59:59	× ೮ * ≣  <b>∨</b> %
Sair		Questionano /	42 /0	1 70	2013-12-13 00.01.00	2013-12-10 20.33.33	* U  <del>*</del>  = **
		Questionário 6	40%	62%	2013-12-06 12:00:00	2013-12-13 16:00:00	×   ぴ   巻   冨   <b>マ</b> め
		Questionário 5	62%	73%	2013-11-19 22:22:00	2013-11-30 23:59:00	×   で   *   置   <b>~</b> め

Figura 1. Tela de gerenciamento de questionários.

Cada questionário oferece a opção de detecção de plágio. Foi criado um identificador simples de plágio, que identifica cópias que alteram somente a formatação e mensagens de texto. A Figura 2 mostra um suposto plágio encontrado entre duas soluções submetidas.

```
Resposta

for i in range(11):
    a = input ('digite um numero: ')
    if a % 2 == 0:
        print 'numero par'
    else:
        print 'numero impar'

for i in range(11):
    a = input ('digite um numero: ')
    if a % 2 == 0:
        print 'numero par'
    else:
        print 'numero impar'
```

Figura 2. Tela de identificação de plágios.

Quando identificado o plágio, o professor ainda precisa confirma-lo já que questões muito simples podem ter soluções realmente parecidas entre os alunos. Sendo assim, fica a critério do professor considerar o plágio. Soluções marcadas como plágio são removidas da avaliação por pares e são marcadas automaticamente como incorretas, sendo assim é recomentado que o processo de marcação dos plágios seja feito antes de liberar o questionário para avaliação dos alunos (que é realizada manualmente).

Aos alunos cadastrados no sistema são apresentados questionários para a avaliação, os questionários anteriormente avaliados com o resultado da avaliação (de cada avaliador) e as questões que ele precisa avaliar.

Para responder um questionário é esperado que o aluno faça uso de um ambiente de programação e apenas copie e cole a solução na ferramenta. Ele pode fazer isso até o momento de encerramento do questionário, podendo inclusive alterar suas soluções até o encerramento. Ao encerramento do questionário o professor deve manualmente liberar o questionário para avaliação através de seu painel, onde são mostrados questionários já encerrados e ainda não liberados para avaliação.

Com relação as questões que o aluno precisa avaliar, é mostrado apenas o enunciado da questão e o algoritmo submetido como resposta. O aluno deverá avaliar o algoritmo a partir de 3 opções: (1) algoritmo correto; (2) algoritmo parcialmente correto; (3) algoritmo incorreto. Caso o algoritmo esteja parcialmente correto ou incorreto, o aluno deverá digitar uma indicação do porque da questão não estar correta. Essa justificativa poderá auxiliar o aluno que elaborou aquela solução a entender seu erro. A Figura 3 mostra a tela onde o aluno deve avaliar uma questão já respondida apresentada para ele (o algoritmo, assim como os demais mostrados nas imagens desse artigo, estão em linguagem Python).



Figura 3. Tela onde o aluno avalia a solução de um colega.

Quando 2 avaliadores fazem a avaliação como correta ou incorreta, o resultado é mostrado com a avaliação final. Quando existe um impasse na avaliação (um avaliador considera a questão correta e o outro incorreta) o professor é quem deve avaliar, resolvendo o impasse. Se ocorrer uma avaliação como correta e outra como parcialmente correta, a questão será considerada correta e se a avaliação como parcialmente correta ocorrer juntamente com uma de incorreta, a questão será considerada incorreta. Duas avaliações como parcialmente correta estão considerando a questão como correta.

A solicitação de uma justificativa para questões consideradas não corretas foi uma melhoria proveniente da reclamação constante na primeira aplicação da ferramenta com relação a falta de retorno sobre o porque de uma questão estar errada ou porque um avaliador considerou errado se outro considerou correto.

A Figura 4 demonstra uma tela onde o aluno pode ver o desempenho de sua solução. Destaca-se a marcação final da questão (como incorreta na primeira questão –

em cor vermelha - e correta na segunda questão - em cor verde) e os comentários de quem avaliou a questão como incorreta ou parcialmente correta. Avaliações feitas como correta aparecem apenas com o texto "Correto" já que não é solicita justificativa do porque o algoritmo está correto.

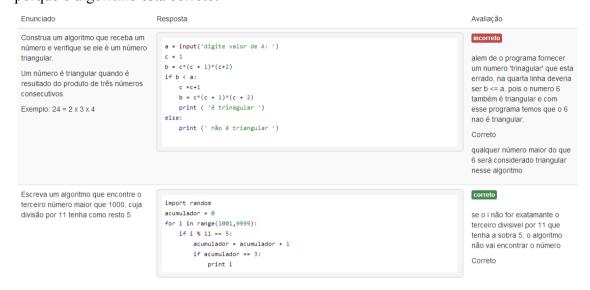


Figura 4. Tela com o resultado das avaliações dos algoritmos submetidos.

Por fim, no processo de avaliação dos alunos, somente questões do último questionário liberado para avaliação ficam disponíveis. O aluno deve manter uma disciplina em responder questões e avaliar questões. Os alunos também mantem um histórico de questões avaliadas e podem visualizar a avaliação feita por outros avaliadores para a questão, podendo comparar seu desempenho como avaliador e permitindo um *feedback* didático. A Figura 5 mostra a tela que ilustra esse recurso. Destaca-se que todas as avaliações são anônimas entre os alunos.



Figura 5. Tela de comparação da avaliação do aluno com os demais colegas.

O professor também pode avaliar as soluções, sendo a avaliação do professor a que possui maior peso. Se os alunos avaliaram como correta a solução e o professor como incorreta, a questão estará incorreta, diminuindo a pontuação dos alunos que avaliaram incorretamente.

Considerar o desempenho do aluno como avaliador, e atribuir uma nota para isso, foi um fator para motivar a avaliação de maneira mais atenciosa e inibir que os alunos apenas avaliem as questões como correta para ajudar os colegas. A Figura 6 mostra parte da tela com a síntese das avaliações do aluno (a esquerda) e o resultado final da avaliação da questão (ao centro). A comparação entre os dois gera a nota (mostrada a direita).

incorreto	incorreto	10
incorreto	соттето	0
parcialmente correto	соттето	10
соггето	correto	10

Figura 6. Tela de comparação e notas das avaliações.

Como pode ser notado na figura anterior, a comparação entre a avaliação do aluno e a avaliação final desconta a nota apenas quando a avaliação do aluno é totalmente contrária a avaliação final, como no segundo item. No terceiro item é visto que o aluno avaliou a questão como parcial e o resultado final foi correto, mas a nota continuou sendo 10. Isso porque para a avaliação final está sendo considerado que duas avaliações como parciais resultam em uma avaliação final como correta, não podendo penalizar o aluno avaliador já que não existe uma avaliação final como parcial.

Como cada questão será avaliada por 2 colegas, isso também define a quantidade de questões que um aluno precisa avaliar. Sendo assim, cada questão respondida pelo aluno exigirá que ele avalie aquela questão para outros 2 colegas. Essa quantidade é configurável, mas é interessante manter como 2 para não tornar o processo de avaliar questões algo penoso por parte do aluno.

Também existe disponível uma opção de avaliação para monitor (ou monitores) da disciplina. O monitor também avalia questões na mesma interface dos alunos, não estando disponível quem submeteu a solução. Não existe limite de questões que um monitor pode avaliar e seu peso é igual ao da avaliação de um aluno.

### 4. Considerações finais

Avaliar uma questão é algo difícil e mesmo tendo uma ferramenta que auxilia no processo, o professor precisa ser um instrumento para melhorar esse aspecto. Muitos alunos avaliam uma solução como incorreta sem uma análise criteriosa, só porque a solução utilizou uma estratégia de resolução diferente da utilizada pelo aluno para resolver. Em um exemplo simplório, para determinar o triplo de um valor o aluno pode multiplicar o valor por 3 ou somar o valor 3 vezes. O aluno que utilizou a primeira solução, espera que os demais também façam a multiplicação, podendo avaliar como incorreta a solução que utiliza as somas. O professor precisa identificar essas situações e comentar em aula para preparar melhor os alunos para serem avaliadores.

Atualmente o aluno precisa codificar em uma ferramenta externa e copiar e colocar a solução na ferramenta apresentada, mas considera-se a possibilidade de adicionar uma IDE online com suporte a debug online, ferramentas que evoluíram substancialmente. Existem diversas soluções de IDEs e debug online de livre acesso e que poderiam ser aplicadas na ferramenta apresentada.

Uma funcionalidade em discussão é a que considera a credibilidade do avaliador na definição do resultado final da questão. A avaliação do aluno com menor credibilidade teria menos peso no resultado final. O desafio está em elaborar uma métrica para isso. Essa solução poderia ser aplicada depois de alguns questionários para evitar o impasse nas avaliações e dispensar o professor de algumas correções.

Algo reclamado pelos alunos é que, uma vez eles tendo avaliado a questão, eles não podem retroceder e alterar a avaliação. Embora essa seja uma questão tecnicamente fácil de implementar, foi mantida por forçar os alunos a pensar mais antes de agir. Mas uma possibilidade derivada disso foi permitir ao aluno avaliado questionar sua avaliação abrindo uma comunicação entre os avaliadores e envolvendo também o professor.

A primeira versão da ferramenta permitia apenas avaliar uma questão como correta ou incorreta. A inclusão da opção de parcialmente correta foi uma solicitação dos alunos e permitiu que pequenos erros pudessem ser menos penalizados. Existem alternativas de avaliação, mas a alternativa escolhida se mostrou satisfatória. Outras formas de avaliação podem ser desenvolvidas e configuradas para uso em uma versão futura.

O identificador de plágio desenvolvido é uma opção bastante simples e limitada. Existem outras opções como o MOSS (Measure of Software Similarity), mas a desenvolvida possui uma vantagem: exercícios simples remetem a respostas muito parecidas, podendo identificadores de plágio resultarem em muitas soluções como plágio. Essa possibilidade de soluções muito parecidas em exercícios simples também foi o motivo de fazer o professor ter que confirmar o plágio, ficando sob sua avaliação a definição final.

Um problema constante é os alunos perderem a data limite de responder o questionário (mesmo que a data fosse sempre no mesmo horário e dia da semana), uma alternativa é enviar e-mail para os alunos que ainda não responderam o questionário, ou o envio de recado no Facebook.

Entre diversos elementos que podem ser avaliados em um algoritmo, o professor pode restringir apenas a geração do resultado correto, embora também possa ser abrangido para a qualidade do código, podendo avaliar o número de variáveis, o significativo nome das variáveis, otimização de estruturas (estruturas de repetição por exemplo), correta indentação do código. Esse critério deve ser apresentado aos alunos e recomenda-se reforça-los de tempos em tempos, conforme desempenho dos alunos na realização das avaliações.

Mesmo que não utilize como critério de avaliação, é esperado que o aluno, corrigindo os algoritmos de colegas, perceba a importância de atribuir nomes significativos para as variáveis, comentar o código em trechos chaves, indentar corretamente, entre outros, pois são elementos que facilitam o entendimento e a manutenção do código.

Ao se deparar com algum recurso desconhecido durante a avaliação de um algoritmo, o aluno pode buscar mais informações sobre aquele recurso a fim de realizar a correta avaliação. Aqui novamente o professor deve assumir o papel de elucidar essa questão da avaliação em sala de aula, pois muitos alunos tendem a avaliar a questão como incorreta quando se deparam com algo que desconhecem e alguns alunos

"inventam" funções que não existem na linguagem, podendo passar a ideia de que o algoritmo está correto.

Atualmente o trabalho está sendo recodificado por questões técnicas. Muitas funcionalidades foram adicionadas e o projeto inicial não pretendia adicionar novas funcionalidades além de ser desenvolvido como protótipo. Uma nova versão será lançada no início do próximo ano.

#### Referencias

- Alves, M. P., et al (2012) "Práticas inovadoras no ensino superior", In: VII Congresso Iberoamericano de Docência Universitária, Porto.
- Barcelos, R. J. S.; Tarouco, L.; Berch, M. (2009) "O uso de mobile learning no ensino de algoritmos", CINTED-UFRGS: Novas Tecnologias na Educação, V. 7 Nº 2.
- Jesus, A. e Brito, G. S. (2010) "Concepção de ensino-aprendizagem de algoritmos e programação de computadores: a prática docente", In: Revista Multidisciplinar da Unioeste, volume 9, numero 16.
- Monteiro, V. e Fragoso, R. (2005) "Avaliação entre pares". In: VIII congresso galaico-português de psicopedagogia, Universidade do Minho, Braga.
- Moreira M. P., Favero, E. L. (2009). "Um Ambiente Para Ensino de Programação comFeedback Automático de Exercícios". In: Workshop Sobre Educação em Computação Anais do XVIII Congresso da Sociedade Brasileira de Computação .Belém-Pará: SBC, 2008.
- Topping, K. (1998), Peer assessment between students in colleges and universities, Review of Educational Research.
- Rodrigues júnior, M. C. (2004) "Experiências Positivas para o Ensino de Algoritmos", II Workshop de Educação em Computação e Informática Bahia-Sergipe, http://www.uefs.br/erbase2004/documentos/weibase/Weibase2004Artigo001.pdf, outubro.