



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
UNIDADE ACADÊMICA DE SISTEMAS E COMPUTAÇÃO  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**IGOR MATHEUS CASTOR DINIZ PINHEIRO**

**AUTOAVALIAÇÃO NA DISCIPLINA DE PROGRAMAÇÃO  
ORIENTADA A OBJETO**

**CAMPINA GRANDE - PB**

**2019**

**IGOR MATHEUS CASTOR DINIZ PINHEIRO**

**AUTOAVALIAÇÃO NA DISCIPLINA DE PROGRAMAÇÃO  
ORIENTADA A OBJETO**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de Bacharel em Ciência  
da Computação.**

**Orientador: Professor Dr. Matheus Gaudencio do Rêgo.**

**CAMPINA GRANDE - PB**

**2019**



P654a Pinheiro, Igor Matheus Castor Diniz.  
Autoavaliação na Disciplina de Programação Orientada  
a Objeto. / Igor Matheus Castor Diniz Pinheiro. - 2019.  
13 f.

Orientador: Prof. Dr. Matheus Gaudencio do Rêgo  
Trabalho de Conclusão de Curso - Artigo (Curso de  
Bacharelado em Ciência da Computação) - Universidade  
Federal de Campina Grande; Centro de Engenharia Elétrica  
e Informática.

1. Disciplina Programação Orientada a Objetos - UFPA.  
2. Autoavaliação. 3. Programação orientada a objetos. 4.  
Avaliação entre pares. I. Rêgo, Matheus Gaudencio do. II.  
Título.

CDU:004(045)

**Elaboração da Ficha Catalográfica:**

Johnny Rodrigues Barbosa  
Bibliotecário-Documentalista  
CRB-15/626

**IGOR MATHEUS CASTOR DINIZ PINHEIRO**

**AUTOAVALIAÇÃO NA DISCIPLINA DE PROGRAMAÇÃO  
ORIENTADA A OBJETO**

**Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.**

**BANCA EXAMINADORA:**

**Professor Dr. Matheus Gaudencio do Rêgo  
Orientador – UASC/CEEI/UFCG**

**Professora Dra. Eliane Cristina de Araújo  
Examinadora – UASC/CEEI/UFCG**

**Professor Dr. Tiago Lima Massoni  
Examinador – UASC/CEEI/UFCG**

**Trabalho aprovado em: 02 de julho de 2019.**

**CAMPINA GRANDE - PB**

# Autoavaliação na Disciplina de Programação Orientada a Objeto

Pinheiro C. D., Igor Matheus

Departamento de Sistemas e Computação, Universidade Federal de Campina Grande (UFCG), Campina Grande, Paraíba, Brasil, [igor.pinheiro@ccc.ufcg.edu.br](mailto:igor.pinheiro@ccc.ufcg.edu.br)

Gaudencio, Matheus

Departamento de Sistemas e Computação, Universidade Federal de Campina Grande, Campina Grande, Paraíba, Brasil, [matheusgr@computacao.ufcg.edu.br](mailto:matheusgr@computacao.ufcg.edu.br)

## ABSTRACT

Feedback is very important for the accretion and fixation of knowledge, but teachers are not always able to provide a timely response to students. Thus, the present work analyzes the applicability of self-assessment as a possible feedback instrument. The experiment counted on 48 Computer Science students who underwent self-evaluation in one of the proposed object-oriented programming course exercises, in the 2019.1 academic year of the Federal University of Campina Grande (UFCG). We found that, with the help of a correction guide, about 66.67% of the students performed an evaluation very close to that of the teacher, which means that the evaluation scores differ by up to 1 point, to verify a correlation between the weighted average of the grades attributed by the students and the final average attributed by the teachers of 0.817, indicating a strong positive relation. In cases of divergence above a point, the students evaluated themselves more rigorously than the teacher of the discipline. The positivity and facilitation of this method in the CS2 course is shown to aid in the reduction of feedback time for students, and it is necessary to carry out further evaluations, for subsequent implementation within the methodologies of the UFCG programming disciplines.

## RESUMO

O feedback é importante para a construção e fixação do conhecimento, mas nem sempre os professores conseguem fornecer rapidamente

informação útil sobre um artefato produzido pelo aluno. Dessa forma o presente estudo analisa a eficiência da autoavaliação como possível instrumento de feedback. O experimento contou com 48 alunos do curso de Ciência da Computação que sujeitaram-se a autoavaliação em um dos exercícios propostos de programação orientada a objetos, da disciplina de Laboratório de Programação 2 (LP2), no período letivo 2019.1 da Universidade Federal de Campina Grande (UFCG). Descobrimos que, com a ajuda de um guia de correção, 32 alunos (cerca de 66,67% da amostra) realizaram uma avaliação bem próxima ao do professor, o que quer dizer que as notas das avaliações diferem em até no máximo 1 ponto, além disso foi possível constatar uma correlação de 0,817 entre a média ponderada das notas atribuída pelos alunos e a média final atribuída pelos professores, indicando uma relação positiva forte. Nos casos de divergência acima de um ponto, os alunos avaliaram-se de forma mais rigorosa que o professor da disciplina. Evidenciando-se a positividade e facilitação desse método na disciplina de LP2 no auxílio a redução do tempo de feedback para os alunos, sendo necessário realização de maiores avaliações, para posterior implantação dentro das metodologias das disciplinas de programação da UFCG.

## KEYWORDS

Ciência da Computação, Programação, Avaliação, Autoavaliação, Programação OO

# 1 INTRODUÇÃO

Os processos avaliativos são de fundamental importância não apenas pelo seu caráter somativo, mas por permitir direcionar o aluno a construção do conhecimento. Isto é ainda mais essencial ao aprendizado de novos paradigmas de programação que ainda é considerado um grande obstáculo durante o ensino de computação (CASPERSEN; KÖLLING, 2009).

Feedback é a informação que o emissor obtém da reação do receptor à sua mensagem, servindo para avaliar os resultados da transmissão e auxiliar no processo de comparação da performance atual com a performance esperada. Atualmente deixou de ser visto apenas como elemento de instrução ou reforço, passando a ser considerado elemento de fundamental importância no processo de aprendizagem, uma vez que há interação da informação fornecida pelo feedback e conhecimento prévio (LIMA; ALVES, 2011).

Dentro desta perspectiva, é possível encontrar um ponto crítico: o fator tempo para obtenção do feedback fornecido pelos professores aos alunos. Esta limitação pode ocorrer devido a muitos fatores, como por exemplo, sobrecarga de trabalho institucional por parte dos professores e quantidade de material a ser avaliado. Em uma situação hipotética, para uma turma de 40 alunos, em que os professores levam 1 hora para concluir um feedback de cada avaliação, haverá demanda de 40 horas de trabalho para conclusão do processo completo de feedback. Se o tempo de resposta for consideravelmente grande, haverá um prejuízo no processo de aprendizado do aluno, e a ferramenta avaliativa deixa de ser um instrumento tão eficaz de aprendizagem.

Sendo assim neste trabalho analisou-se a eficácia e eficiência da aplicação do processo de autoavaliação com os alunos da disciplina de Laboratório de Programação II (LP2) da UFCG (Universidade Federal de Campina Grande). Foram analisados os resultados das autoavaliações dos alunos matriculados na disciplina anteriormente citada, no período 2019.1, sobre suas próprias provas, e comparados com os resultados fornecidos pelos professores, com a finalidade de verificar se o processo de autoavaliação é um instrumento ativo no combate do desafio relacionado à redução do tempo de feedback e se a mesma se mostra eficiente dentro do escopo desta disciplina.

Participaram do processo de autoavaliação 48 alunos, que guiados por diretrizes de correção fornecidas pelos professores, atribuíram médias para si para cada categoria solicitada de um exercício avaliativo. Como resultado, 32 alunos (cerca de 66,67%) que participaram da autoavaliação atribuem uma nota com, no máximo, 1 ponto de diferença da nota atribuída pelo professor. E o coeficiente de correlação entre a média das notas atribuídas pelos alunos e a média das notas atribuídas pelos professores é de 0,817, indicando uma relação forte positiva.

A metodologia da pesquisa proposta neste trabalho é exibida na Seção 3 para em seguida apresentar os resultados obtidos. Na Seção 2, discutiremos a fundamentação teórica e trabalhos relacionados e apresentamos, ao final do trabalho, nossas conclusões.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Avaliação Entre-pares e Autoavaliação

Já foram explorados, em alguns casos e estudos, correções automáticas para a rápida geração de feedback (GAUDÊNCIO et al., 2013) entretanto tal instrumento se adaptou de melhor maneira às disciplinas mais introdutórias de ensino de programação, isso ocorre porque em níveis introdutórios existe uma análise com enfoque em aspectos funcionais e portanto passíveis de automação. Outra abordagem pesquisada para melhorar o tempo de resposta foi o processo de avaliação entre pares, que mostrou um alto índice de semelhança com a avaliação fornecida pelos professores (ARAÚJO; GAUDÊNCIO, 2018)

A avaliação entre pares confere ao aluno um nível de investigação em relação ao material produzido pelo colega, gerando senso de análise crítica, reflexiva e maior nível de aprendizado, porém segundo Cartney (2010), este processo está sujeito a fatores emocionais, onde os avaliadores se sentem ansiosos e preocupados em ser rígido demais com o colega ou fatores relacionados a estes.

A autoavaliação por sua vez é a habilidade de analisar-se criticamente e segundo Boud (2013) tem um papel muito forte no que diz respeito à fixação de conhecimentos. Além disso confere ao aluno um caráter mais pró-ativo em seu processo de aprendizado (EARL, 2003). Entretanto não

existem muitos trabalhos que comprovem ou confirmem a aplicação desta metodologia no meio computacional (FRANÇA; TESDESCO, 2015).

## 2.2 Trabalhos relacionados

O processo de avaliação tem passado por várias mudanças para melhor adaptar-se ao contexto acadêmico. A mudança principal diz respeito à forma como a avaliação é compreendida e assimilada pela parte avaliada, e que segundo Cartney (2010) caso haja uma má assimilação o processo de interiorização e absorção de conhecimento é comprometido. Em seu trabalho de pesquisa-ação optou pela metodologia de avaliação entre pares para entender se a mesma pode ser utilizada como veículo de aprendizado para os alunos. Foi possível perceber que avaliação entre pares é melhor aplicada em grupos pequenos, enquanto que a questão de gerência do grupo se torna quase que impraticável se o mesmo for muito grande.

Cartney (2010) avalia ainda que a maioria dos alunos se mostraram ansiosos em assumir o papel de avaliador para outro aluno, temendo ser rígido em demasia ou se auto julgar não bom o suficiente para assumir tal papel. O lado avaliado também demonstrou-se afetado pelo lado emocional, ficando ansiosos e preocupados por serem expostos a colegas de classes no qual passam a maior parte do tempo, o que difere dos resultados obtidos pela autoavaliação, uma vez que os alunos julgam a si mesmos, não expressando o sentimento de exposição para outrem.

No trabalho de Sitthiworachart e Joy (2003) desenvolveu-se um ambiente web para que os alunos registrassem o resultado das avaliações entre pares, uma plataforma própria para testar a eficácia desta metodologia na aprendizagem em linguagens de programação. Em nosso caso utilizamos uma plataforma digital pronta, o Canvas Instructure, com enfoque na parte de autoavaliação;

Foi realizado ainda com o intuito de analisar a viabilização da aplicação da avaliação entre pares na mesma disciplina de programação com paradigma OO. Segundo Araújo e Gaudêncio (2018), o mecanismo de avaliação entre pares é um mecanismo útil na medida que coloca o aluno no papel de revisar a atividade do colega e de ser capaz de gerar, rapidamente, algum tipo de feedback. Percebe-se portanto que existe um ponto de concordância entre a avaliação entre pares e a

autoavaliação, em ambos os casos existe um olhar diferenciado a respeito do aluno, conferindo um papel mais proativo no processo de aprendizado. Todo esse processo pode ser considerado uma metodologia ativa, uma vez que busca aperfeiçoar a autonomia individual do aluno e, “se queremos que os alunos sejam proativos, precisamos adotar metodologias em que os alunos se envolvam em atividades cada vez mais complexas [...]”, (MORÁN, 2015, p. 17).

O instrumento de pesquisa deste trabalho foi semelhante ao utilizado por Araújo e Gaudêncio (2018), entretanto algumas modificações foram feitas para minimizar os desvios relacionados às categorizações das notas e possíveis interpretações erradas da parte dos alunos avaliadores. Foi entregue aos alunos um guia avaliativo, como uma forma de contribuir positivamente na autoavaliação.

Segundo Hsiao, Brusilovsky e Sosnovsky (2008), às perguntas parametrizadas contribuem no processo de aprendizagem. No respectivo estudo, 31 alunos foram submetidos ao uso da ferramenta QuizJET, como instrumento de autoavaliação. A mesma apresenta perguntas parametrizadas e leva o aluno a resolução, para contribuir no aprendizado. Percebeu-se que 87,5% dos estudantes aprovaram o uso do QuizJET, comprovando a eficácia das perguntas parametrizadas. Entretanto, a ferramenta utilizada não permite a autoavaliação de algo já produzido pelos alunos, sendo o seu foco a fixação de conteúdo OO. Em nosso estudo, o quiz de autoavaliação tem enfoque em códigos produzidos pelos próprios alunos.

## 3 METODOLOGIA

É comum que o primeiro contato de muitos alunos com o paradigma de programação Orientada a objetos aconteça na disciplina de Laboratório de Programação 2. Para este grupo de alunos, o objetivo da disciplina é transmitir de maneira eficaz os conceitos de OO, de modo que possam se sentir seguros em momentos futuros ao se depararem com situações que requerem o uso deste paradigma. Durante a disciplina os alunos são apresentados a alguns tópicos específicos como modularização de programas, uso de objetos, criação de classes, encapsulamento, herança e composição, polimorfismo, tipos abstratos de dados, tipos de coleções de objetos, tratamento de erros com

exceções, testes de unidade, regras básicas de design e refatoramento.

Assim como qualquer disciplina acadêmica, LP2 possui sua própria dinâmica. Durante o processo avaliativo os alunos recebem especificações de “mini-projetos” e avaliações, que se assemelham a alguma situação do mundo real, com um número consideravelmente pequeno de funcionalidades a serem implementadas. Para realizar a implementação das atividades solicitadas os alunos utilizam de conceitos vistos em sala de aula para cobrir toda a especificação fornecida.

Para a realização desta pesquisa, contamos com a participação de 48 graduandos avaliando um de seus próprios exercícios avaliativos realizados na aula anterior. Este exercício avaliativo é composto por uma pequena descrição de uma situação do mundo real, e alguns passos a serem implementados. Cada passo, possui um objetivo de avaliação específico, tendo ligação direta com a situação problema descrita. Inserido dentro deste contexto, o aluno deve implementar a solução solicitada em cada passo, utilizando os conhecimentos de OO, atendendo a lógica de funcionamento desejada para cada passo.

Para o processo de autoavaliação, todos os alunos recebem um guia de correção indicando

como avaliar seus exercícios, como mostra a Tabela 1. O mesmo foi elaborado a partir de discussões entre a equipe de professores de Programação II sobre quais aspectos deveriam ser percebidos e estudados nas avaliações dos alunos. Buscando ser o mais claro possível para que os alunos não se sentissem inseguros com o instrumento de pesquisa em mãos, o guia de correção era composto por 5 questões: 2 questões para avaliação das classes básicas a serem implementadas, 1 questão para avaliação dos testes elaborados, 1 questão sobre comentários gerais a respeito do código ou da prova e 1 questão sobre uma possível nota atribuída pelo aluno para sua avaliação. Ainda sobre o guia de correção, as questões referentes à avaliação das classes básicas possuíam um conjunto de três critérios, sendo eles, avaliação dos construtores, atributos e métodos. A ferramenta foi pensada de uma forma que qualquer professor de Programação II consiga aplicá-la a sua equipe de alunos obtendo os resultados semelhantes e satisfatórios. A Tabela 1 é um resumo do guia avaliativo, nos anexos é possível encontrar a tabela completa com os critérios utilizados.

**Tabela 1: Resumo do guia avaliativo repassado para que os alunos se auto avaliassem**

Questão	O aluno deve identificar...	Cálculo da nota da questão...
Quanto a classe Delivery, veja se o código atende os seguintes requisitos:	<p>ATRIBUTOS (encapsulamento, visibilidade e tipo adequados)</p> <p>CONSTRUTOR (parâmetros adequados e inicialização)</p> <p>MÉTODOS implementados corretamente.</p>	<p>Atributos = 2 - 0,5(penalização)</p> <p>Construtor = 2 - 0,5(penalização)</p> <p>Métodos = 6 - 2(penalização)</p>
Quanto a classe DOD, veja se o código atende os seguintes requisitos:	<p>ATRIBUTOS (encapsulamento, visibilidade e tipo adequados)</p> <p>CONSTRUTOR (parâmetros adequados e inicialização)</p> <p>MÉTODOS implementados corretamente.</p>	<p>Atributos = 1 - 0,5(penalização)</p> <p>Construtor = 2 - 0,5(penalização)</p> <p>Métodos = 7 - penalização do método infringido</p>
Quanto aos testes do código	Se existiam todos os testes solicitados	Teste = 10 - penalização do teste ausente

Faça um comentário sobre a prova avaliada. Sinta-se à vontade para escrever uma recomendação ao responsável pela prova, apresentando pontos positivos e negativos.	Comentário pessoal.	Sem cálculo de nota.
--	---------------------	----------------------

## 4 RESULTADOS E DISCUSSÕES

A partir da aplicação do instrumento de pesquisa mostrado anteriormente, obteve-se a coleta de dados de 48 alunos que se autoavaliaram. Em seguida com as notas das médias atribuídas pelo aluno para cada categoria, foi calculado uma média ponderada, levando em consideração pesos específicos do escopo da disciplina, sendo 30% da nota da final da classe Delivery, 50% da nota final para a classe Delivery Of the Day (DoD) e 20% para a nota final de testes. Da mesma forma os professores também avaliaram os alunos, para as mesmas categorias, finalizando esse processo com o mesmo cálculo da média ponderada. Logo em seguida foram comparadas as médias dos alunos e as médias dos professores.

Na Figura 1, é possível perceber que a maioria dos alunos gasta até 40 minutos para realizar a autoavaliação, isto mostra que este procedimento é aplicável dentro do escopo da disciplina, uma vez que possui uma carga horária de 2 horas/aula. O aluno que mais demorou a terminar o processo utilizou 105 min, o que ainda está dentro das 2 horas/aula. Mesmo com a prática, os professores reportaram um tempo entre 10 a 20 minutos para correção por prova, o que implica em, pelo menos, 8 horas para a realização da correção.

Na Figura 2 é possível perceber que na maioria dos casos, a diferença entre as médias das autoavaliações e as médias dos professores foi entre -1,17 a 1,17 pontos, 32 dos 48 alunos tiveram uma diferença inferior ou igual a 1 ponto, representando 66,67% da nossa amostra. 18 alunos se auto avaliaram de forma mais rigorosa, ou seja deram uma nota mais baixa que a do professor para si e 27 alunos se auto avaliaram de forma menos rigorosa, dando para si uma nota um pouco mais alta que a do professor. A média das diferenças entre as médias das notas do professor e do aluno foi de 0,7 e a mediana 0,87.

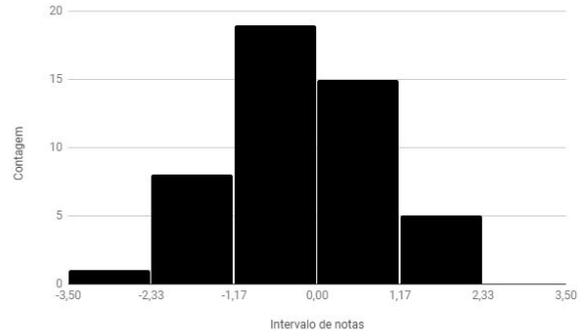
Foi possível perceber que para tais casos, os alunos foram mais rígidos nos quesitos funcionalidades e testes, penalizando com

severidade situações em que os testes automáticos falharam, ou até mesmo anulando toda a pontuação caso o teste implementado não fosse exatamente igual ao solicitado. Funcionalidades não implementadas da forma solicitada também foram severamente punidas pelos próprios alunos, situações em que os professores procuram ser mais compreensivos, analisando se existem casos em que os testes automáticos sejam bem sucedidos, dentro do escopo solicitado na prova ou se a funcionalidade, está de algum modo semelhante ao solicitado. Avaliações da classe Delivery por partes dos alunos foram as avaliações que mais se aproximaram avaliação dos professores, deve-se ao fato de ser uma classe uma mais simples, com menos funcionalidades e métodos.

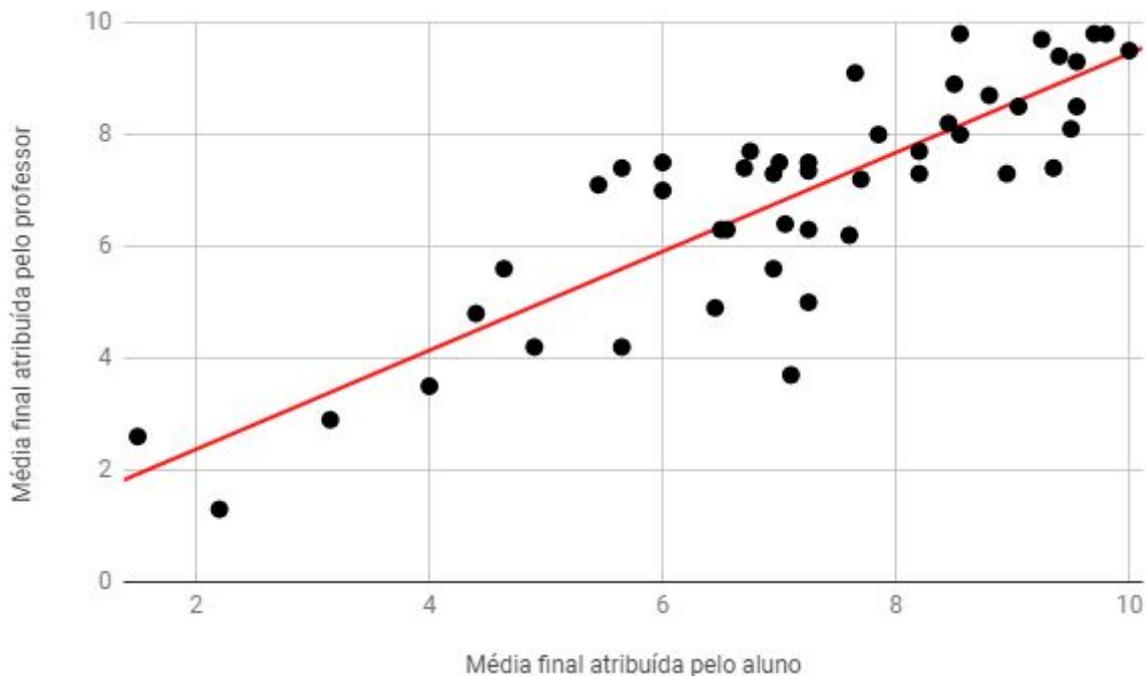
Para fortalecer a ideia de que realmente existe uma relação entre as médias dos alunos auto avaliados e as médias dos professores, realizou-se um teste estatístico de efeito. Tendo em vista que notas de provas podem ir de 0 a 10, que pode caracterizar uma faixa de valores não normalizados, se fez necessário ranquear todas as notas, para subsequentemente aplicar um teste de correlação. Com as notas ranqueadas, ao aplicar o teste de correlação obtemos um valor para o coeficiente de correlação de 0,817. Na Figura 3 é possível observamos o gráfico de dispersão das variáveis em estudo, nele é possível perceber que existe de fato uma linha de tendência linear crescente. O resultado para o coeficiente de correlação obtido para este experimento está bem próximo de 1, refletindo uma relação positiva bastante forte, o que quer dizer que se um aluno se autoavalia com uma nota alta, é bem provável que o professor também irá o avaliar com uma nota alta, considerando uma margem de erro.



**Figura 1: Gráfico do tempo gasto pelos alunos para realizar a auto avaliação**



**Figura 2: Diferença de notas entre Professores e Alunos**



**Figura 3: Gráfico de dispersão para as variáveis Nota final atribuída pelos alunos x Nota final atribuída pelos professores tempo gasto pelos alunos**

## 5 CONCLUSÃO

A autoavaliação é um processo que estimula o aprendizado do aluno, colocando-o num papel ativo ao avaliar sua própria atividade. O resultado gerado pela autoavaliação tem impacto direto no ponto de gargalo neste trabalho mencionado, o tempo de resposta por parte dos professores, que chamamos de feedback. Uma vez com posse de seu próprio feedback orientado por um guia de correção fornecido pelos professores, os alunos terão consciência se estão trilhando o caminho correto. Neste trabalho foi possível

observar uma forte correlação (0,817) entre as notas dos alunos para si e as notas dos professores, além disso apenas 16 alunos tiveram a diferença da média atribuída por si mesmo em comparação a média do professor diferindo por mais de um ponto, a média das diferenças foi de 0,7 e mediana de 0,87, refletindo que o feedback por autoavaliação é algo realmente válido e que pode ser aplicado de forma eficaz na disciplina de Laboratório de Programação 2, como forma de auxílio aos professores. Ainda foi possível observar que os alunos são mais rígidos em alguns pontos como funcionalidades e testes, mas que apresentam argumentos bastante similares aos professores com

relação a critérios como design, categoria esta que não teve uma diferença de notas significativa entre os alunos e os professores.

Nesta perspectiva, a autoavaliação permite que alunos tenham um feedback bem mais rápido do que o convencional, agilizando o processo de aprendizado. Além do caráter de auxílio aos professores no quesito de avaliação somativa, quando bem guiados pelo instrumento de pesquisa construído pela equipe de professores da disciplina, os alunos constroem uma avaliação formativa consistente, uma vez que os parâmetros formais desejados estão explicitados no guia avaliativo.

Como trabalhos futuros, pretende-se replicar o experimento contando com um número maior de alunos, acompanhando-os durante todo o período letivo, analisando se a autoavaliação tem impacto considerável sobre o desempenho do aluno ao final da disciplina. Aplicar o experimento também para disciplinas além de Programação II, permitirá o entendimento de como a autoavaliação pode auxiliar no processo de interiorização do conhecimento. Por fim, um estudo qualitativo com enfoque e olhar crítico sobre os pontos positivos e negativos da autoavaliação, apontados pelos alunos, pode ser uma alternativa para aperfeiçoar o instrumento, e melhorar o tempo de feedback para os estudantes.

## REFERÊNCIAS

- ARAÚJO, V.B.; GAUDENCIO, M. Avaliação Entre-Pares na Disciplina de Programação OO. In: VII CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2018. **Anais: 10.5753/cbie.sbie.2018.1955**. Disponível em: <http://br-ie.org/pub/index.php/sbie/article/view/819>. Acesso em fev. 2019.
- BOUD, D. **Enhancing Learning Through Self-assessment**. 1.Ed. London: Routledge, 2013.
- CARTNEY, P. Exploring the use of peer assessment as a vehicle for closing the gap between feedback given and feedback used. **Assessment & Evaluation in Higher Education**. v.35,n.5, p.551–564, August, 2010.
- CASPERSEN, M.E.; KOLLING, M. Stream: A First Programming Process. **ACM Transactions on Computing Education (TOCE)**, Volume 9 Issue 1, Article N. 4. march. 2019.
- EARL, L.M. **Assessment as Learning: Using Classroom Assessment to Maximize Student Learning**. 2.Ed. California: Corwin, 2013.
- FRANÇA, R.S.; TESDESCO, P.C.A.R. Caracterizando a pesquisa sobre autoavaliação na aprendizagem de programação para iniciantes. In: XXVI SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2015. **Anais: 10.5753/cbie.sbie.2015.549**. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/5312>. Acesso em fev. 2019.
- GAUDENCIO, M., et al. Eu Sei o que Vocês Fizeram (Agora e) na Aula Passada: o TSTView no Acompanhamento de Exercícios de Programação. In: II CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2013. **Anais: 10.5753/CBIE.SBIE.2013.204**. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/2498>. Acesso em fev. 2019.
- HSIAO, I.H.; BRUSILOVSKY, P.; SOSNOVSKY, S. Web-based Parameterized Questions for Object-Oriented Programming. **School of Information Sciences**, University of Pittsburgh. Pittsburgh, United States, 2008.
- LIMA, D.M.A.; ALVES, M.N. O feedback e sua importância no processo de tutoria a distância. **Pro-Posições**. v.22,n.2, p.189-205, Maio/ago, 2011.
- MORAN, J. **Mudando a educação com metodologias ativas**. v.II. Coleção Mídias Contemporâneas. Convergências Midiáticas, Educação e Cidadania: aproximações jovens, 2015.
- SITTHIWORACHART, J.; JOY, M. Web-based Peer Assessment in Learning Computer Programming. In: THE 3rd IEEE INTERNATIONAL CONFERENCE ON ADVANCED LEARNING TECHNOLOGIES, 2003. **Anais:10.1109/ICALT.2003.1215052**. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/2498>. Acesso em fev. 2019. Acesso em fev. 2019.



# APÊNDICES

Guia avaliativo da tabela 1

Questão	O aluno deve identificar...	Cálculo da nota da questão...
<p>Quanto a classe Delivery, veja se o código atende os seguintes requisitos:</p>	<p>- Quanto aos ATRIBUTOS (2.0): Os mesmos estavam com encapsulamento, visibilidade e tipo adequados</p> <p>- Quanto ao CONSTRUTOR (2.0): Recebia os parâmetros adequados e inicializava corretamente os atributos do objeto criado com os parâmetros recebidos Inicializava o status para ready</p> <p>- Quanto aos MÉTODOS (6.0) Os mesmos implementavam corretamente a funcionalidade proposta:</p> <p>setEstado (2.0) equals/equals (2.0) toString (2.0)</p>	<p>Atributos = 2 - 0,5(penalização) Construtor = 2 - 0,5(penalização) Métodos = 6 - 2(penalização)</p>
<p>Quanto a classe DOD, veja se o código atende os seguintes requisitos:</p>	<p>- Quanto aos ATRIBUTOS (1.0): Os mesmos estavam com encapsulamento, visibilidade e tipo adequados</p> <p>- Quanto ao CONSTRUTOR (2.0): Oferecia construtor com número padrão de entregas e recebendo o array/coleção de clientes Oferecia construtor com limite personalizado de entregas e recebendo o array/coleção de clientes</p> <p>- Quanto aos MÉTODOS (7.0): Os mesmos implementavam corretamente as funcionalidades propostas:</p> <p>addDelivery - altera estado do delivery (1.0) addDelivery - não adiciona delivery existente (1.0) addDelivery - checa se o cliente é válido (0.5) addDelivery - checa limite de</p>	<p>Atributos = 1 - 0,5(penalização) Construtor = 2 - 0,5(penalização) Métodos = 7 - penalização do método infringido</p>

	entregas (0.5) listDoD - na ordem inserida (1.0) listFinished - na ordem inserida (1.0) finishDelivery - verifica pos invalida (1.0) finishDelivery - finaliza deliveries (1.0)	
Quanto aos testes do código	construtor padrão - 6 entregas (0.5) construtor c/ limite de entregas (0.5) addDelivery - altera estado do delivery (1.5) addDelivery - não adiciona delivery existente (1.5) addDelivery - checa se o cliente é válido (1.0) addDelivery - checa limite de entregas (1.0) finishDelivery - verifica posição inválida (2.0) finishDelivery - finaliza Delivery (2.0)	Teste = 10 - penalização do teste ausente
Faça um comentário geral sobre a prova avaliada. Sinta-se a vontade para escrever uma recomendação ao responsável pela prova, apresentando pontos positivos, negativos e comentários que achar interessante.	Comentário pessoal.	Sem cálculo de nota.