Os estudantes leem as mensagens de feedback estendido exibidas em juízes online?

Joseph de Oliveira[1], **Felipe A. Salem**[1], **Elaine H. T. Oliveira**[1], **David B. F. Oliveira** [1], **Leandro S. G. Carvalho**[1], **Filipe Dwan Pereira**[2]

[1]Instituto de Computação – Universidade Federal Amazonas (UFAM)

[2]Departamento de Ciência da Computação – Universidade Federal de Roraima (UFRR)

{jvlo,david,galvao,elaine}@icomp.ufam.edu.br, filipe.dwan@ufrr.br

Resumo. *Apesar dos benefícios dos juízes online em relação ao processo de correção automática de códigos, a literatura afirma que o feedback desses sistemas precisa ser aprimorado para gerar um aprendizado mais efetivo para o estudante. Este artigo aprofunda um estudo anterior, do tipo intervenção-controle, e visa identificar se o feedback estendido apresentado aos alunos de programação introdutória foi visualizado, e se houve diferença estatisticamente significante entre o interesse dos diferentes grupos pelo conteúdo do feedback. Foi observada diferença entre os alunos dos grupos teste e controle em relação ao tempo de leitura, tempo de leitura por palavra, ocorrências de alguns tipos de erro e e em relação à frequência de acesso ao feedback estendido.*

Abstract. *Despite the benefits of online judges over the automatic code correction process, the literature states that feedback from these systems needs to be improved to generate more effective learning for the student. This article deepens an earlier study, intervention-control type, and aims to identify whether the extended feedback presented to the students of introductory programming was visualized, and whether there was a statistically significant difference between the interest of different groups in the content of the feedback. A difference was observed between students in the test and control groups in relation to reading time, reading time per word, occurrences of some types of errors and and in relation to the frequency of access to extended feedback.*

1. Introdução

Uma das maiores dificuldades apresentadas por estudantes de disciplinas de programação introdutória é entender o significado das mensagens de erro de compilação, conhecidas como CEM (*Compiler Error Message*) [Becker et al. 2016]. Além da barreira com o idioma inglês, os estudantes podem não compreender qual o erro apontado pela mensagem, pois as CEMs normalmente são escritas tendo programadores como público-alvo, e não aprendizes. Além disso, as CEMs apontam apenas qual a situação de erro, mas não dão nenhuma dica ao estudante sobre o que pode ser feito para contornar o problema.

Este artigo aprofunda a análise realizada por nós em um trabalho anterior, [de Oliveira et al. 2019], em que foi investigada a influência das mensagens de erro apresentadas a estudantes de introdução à programação sobre seus desempenhos na disciplina. Nesse trabalho anterior, que iremos chamar de artigo de referência, os alunos envolvidos na pesquisa resolviam os exercícios da disciplina usando a linguagem Python, com o apoio de um ambiente de correção automática de códigos (ACAC), também conhecido

DOI: 10.5753/cbie.sbie.2020.1723

como juiz online. Durante a pesquisa, os estudantes foram divididos aleatoriamente em dois grupos. Ao testar ou submeter um código com erro, os estudantes do grupo *teste* recebiam uma tradução para o português da mensagem de erro do compilador Python e também algumas dicas de como resolvê-lo. Já os alunos do grupo *controle* recebiam apenas a tradução da mensagem de erro.

O trabalho anterior concluiu que os alunos que receberam o novo tratamento testaram mais vezes o código, porém não foi realizada uma análise sobre a leitura do feedback criado. O presente trabalho, por sua vez, vai além da questão do desempenho e investiga a receptividade e utilidade do feedback por parte dos estudantes, com indicadores relacionados ao tempo de leitura, cometimento de erros e à percepção dos alunos sobre a utilidade do feedback criado. Portanto, este trabalho traz uma contribuição em relação à avaliação de método proposto por [de Oliveira et al. 2019], algo crucial para análise da validade do método. Para tanto, as seguintes questões de pesquisa foram levantadas:

Q1: Há diferença estatisticamente significante no tempo de leitura do feedback entre os grupos?
Q2: Há diferença estatisticamente significante na densidade de leitura do feedback entre os grupos?
Q3: Há diferença estatisticamente significante na quantidade de visualizações do feedback entre os grupos?
Q4: Há diferença estatisticamente significante na quantidade de erros cometidos entre os grupos?
Q5: Há diferença estatisticamente significante em como os alunos de cada grupo percebem a utilidade das mensagens de feedback?

2. Trabalhos relacionados

As mensagens de erro do compilador, chamadas de *Compiler Error Messages* (CEMs), são alguns dos recursos essenciais que as linguagens oferecem aos programadores. Contudo, elas geralmente são difíceis de interpretar e representam uma barreira para os alunos iniciantes em programação [Becker et al. 2016]. Com isso, [Nienaltowski et al. 2008] propuseram o uso das mensagens aprimoradas de erro do compilador, chamadas de *Enhanced Compiler Error Messages* (ECEMs), que abrangem, além das mesmas informações que as CEMs, um feedback sobre o que fazer para consertar o erro.

Feedback são as informações pós-resposta que são fornecidas ao aluno para informá-lo de seu estado real de aprendizado ou desempenho [Narciss 2008]. Com base nesse conceito, [Keuning et al. 2018] classificaram o feedback em três tipos: 1) conhecimento sobre restrições de tarefas; 2) conhecimento de resultado/resposta; 3) conhecimento de desempenho para um conjunto de tarefas. O presente estudo situa-se nos tipos 2 e 3. Pois o ACAC utilizado inspeciona os códigos dos alunos e gera uma avaliação instantânea sobre a corretude da solução com base em casos de teste. Além disso, informa aos alunos o seu desempenho em cada conjunto de tarefas resolvido pelo aluno.

Para verificar a efetividade do feedback para a linguagem Java, [Becker 2016] tomou como base o número total de erros do aluno, e [Nienaltowski et al. 2008], o percentual de acertos. A conclusão de [Nienaltowski et al. 2008] foi que não havia diferença estatisticamente significante no comportamento dos alunos. Em contraste, [Becker 2016] aprimorou de maneira semelhante as CEMs e suas descobertas mostraram que as ECEMs realmente mudam o comportamento dos alunos, visto que houve uma redução no número total de erros dos alunos que as receberam.

A vantagem das ECEMs é amplamente compreendida, visto que alunos que as recebem relatam ter um ótimo feedback, enquanto um grupo de alunos que só recebe CEMs reportaram a ausência de informação [Denny et al. 2020]. [Karvelas et al. 2020] compararam o comportamento dos programadores usando 2 versões do Blue J, um ambiente de desenvolvimento integrado para desenvolvimento Java. Com isso, foi observado que alunos obtiveram melhor desempenho na versão em que as CEMs possuem maior carga cognitiva. [Prather et al. 2017] testaram a compreensão dos alunos das ECEMs e demonstraram sua eficácia em relação as CEMs, mesmo contendo uma carga cognitiva maior. Os resultados confirmam os de [Karvelas et al. 2020], pois em ambos houve melhorias no desempenho dos alunos, mesmo tendo uma carga cognitiva maior, mais da metade dos alunos, em ambos os estudos, tiveram compilações bem sucedidas.

[Jesus et al. 2018] e [Alves and Jaques 2014] encontraram, por meio de avaliações em ACACs, diferenças significativas na utilização das ECEMs. A avaliação das ECEMs, feita pelo aluno em ambos os trabalhos, demonstra grande utilidade visto que foram observadas melhorias no desempenho do aluno em termos de média final e tempo de programação. Porém as pesquisas contaram com poucos participantes: 26 e 24 participantes, respectivamente, enquanto o presente trabalho conta com 179 participantes.

Os trabalhos anteriores investigaram o impacto das ECEMs no aprendizado de programadores iniciantes, e observaram que mesmo contendo uma carga cognitiva maior, elas são de grande utilidade para os estudantes. Já este trabalho calcula métricas relacionadas à leitura das ECEMs como o tempo de leitura e a densidade de leitura do feedback e com isso busca investigar o impacto delas no comportamento da depuração do código.

3. Metodologia

Esta seção apresenta a metodologia utilizada para comparar a visualização de mensagens, tempo de leitura e utilidade do feedback estendido nos grupos controle e teste.

3.1. Contexto

As mensagens de feedback foram implementadas no ACAC Codebench, utilizado na Universidade Federal do Amazonas, para apoiar o aprendizado e a avaliação de 17 turmas de introdução à programação de computadores (IPC), ofertadas a cursos de graduação em ciências exatas e engenharia. O ACAC utilizado sofreu grandes mudanças entre os anos de 2018 e 2019. Para contornar essa ameaça à validade, adotamos um desenho experimental de comparação entre grupos de teste e controle e utilizamos 5 turmas de IPC para a análise. Os estudantes foram estratificados apenas por sexo, para evitar problemas de balanceamento nos dados. Além disso, realizamos a distribuição dos alunos entre as turmas que historicamente apresentam altas e baixas taxas de aprovação, para contornar os vieses de participação nas atividades e de aprovação na disciplina.

Entre as 5 turmas de IPC em 2019, o experimento iniciou com 274 alunos matriculados, dos quais apenas 179 permaneceram ativos. Para classificar os alunos como ativos ou desistentes, aplicamos um questionário na metade do semestre letivo, de modo que seu preenchimento era pré-requisito para acessar as atividades seguintes do curso. Dessa forma, identificamos os alunos desistentes e analisamos a receptividade às ECEMs, por meio das respostas ao questionário, descritas na Seção 4.5. Os dados sobre os alunos incluídos na análise estão dispostos na Tabela 1.

Conforme apresentado no artigo referência, identificamos os erros mais cometidos pelos estudantes nas turmas anteriores da disciplina e criamos ECEMs para cada um deles.

Tabela 1. Caracterização dos alunos participantes do experimento

	Característica			
Grupo	**Homens**	**Mulheres**	**Total**	**Idade (anos)**
Teste	57 (69,5%)	25 (30,5%)	82	20,4 $\pm$ 3,83
Controle	67 (69,1%)	30 (30,9%)	97	20,4 $\pm$ 4,53
Total	124 (69,3%)	55 (30,7%)	179	20,4 $\pm$ 4,21

Juntos, esses tipos corresponderam a 79,61% dos erros cometidos pelos estudantes entre 2016 e 2018. Não criamos ECEMs para outros tipos de erros, pois sua elaboração e validação consomem tempo e elas são pouco frequentes (menos de 1%).

Para os estudantes do **grupo teste**, o ACAC exibe dois tipos de informação quando o compilador Python identifica um dos erros frequentes (Tabela 4): a **mensagem de erro traduzida para o português** e a **mensagem de feedback estendido**. Para os estudantes do **grupo controle**, o ACAC exibe apenas a **mensagem traduzida**, sem o feedback estendido. Essa configuração foi mantida nos 4 primeiros módulos da disciplina (primeira metade do período letivo). Nos 3 últimos módulos, o ACAC foi programado para inverter a regra de exibição de mensagens de feedback, de forma que o grupo controle passou a receber o feedback estendido e o grupo teste passou a receber apenas as traduções dos erros. Essa inversão foi feita para garantir a integridade ética da pesquisa, a fim de que todos os estudantes tivessem acesso aos mesmos conteúdos, ainda que em tempos diferentes.

3.2. Novas métricas de análise

Partindo dos mesmos dados utilizados no artigo referência, estabelecemos novas métricas de análise, desta vez relacionadas à interação dos estudantes com as mensagens de feedback (Tabela 2). Por meio delas, deseja-se compreender a relevância das mensagens de feedback para os estudantes por duas vias: (i) rastros de abertura e visualização das mensagens de feedback, e (ii) respostas dos estudantes a um questionário de opinião.

Tabela 2. Métricas observadas neste estudo, coletadas a partir do ACAC

Métrica	**Significado**
feedback_aberto	variável booleana que sinaliza se a mensagem de feedback foi aberta pelo aluno
erro_quantidade	quantidade de erros cometidos pelo aluno
tempo_leitura	tempo de leitura, em segundos, de uma mensagem de feedback
densidade_leitura	tempo de leitura dividido pelo número de caracteres do feedback

4. Resultados

Nesta seção, são apresentados os resultados encontrados para a metodologia proposta, estruturados conforme as questões de pesquisa apresentadas na Introdução. Foi realizado o teste de Shapiro-Wilk para verificar a normalidade dos dados, e constatou-se que nenhum dos parâmetros avaliados seguia a distribuição normal. Em seguida, por termos duas amostras independentes em cada análise, relativas aos grupos de tratamento, aplicamos o teste não paramétrico de Mann-Whitney para verificar se havia diferença significativa entre os grupos teste e controle, com um nível de confiança de 95%.

4.1. Q1: Há diferença estatisticamente significante no tempo de leitura do feedback entre os grupos?

Na Figura 1, podemos observar que o grupo teste obteve um tempo de leitura maior do que o grupo controle. Também observamos no grupo teste um terceiro quartil e limite superior com valores maiores do que os do grupo controle, indicando uma possível preferência pelo conteúdo do feedback estendido. Analisando a razão entre total do tempo de leitura do feedback e o número de alunos em cada grupo, obtivemos um valor de 317,20 segundos para os alunos do grupo teste e 102,30 segundos para os alunos do grupo controle, indicando que os alunos do grupo teste também leram por mais tempo o conteúdo quando levamos em consideração os tamanhos de cada grupo. Também foi encontrada diferença estatisticamente significante entre os dados dos grupos, com um $p < 0{,}0001$ para o atributo de tempo de leitura, indicando que os dados pertencem a distribuições diferentes.

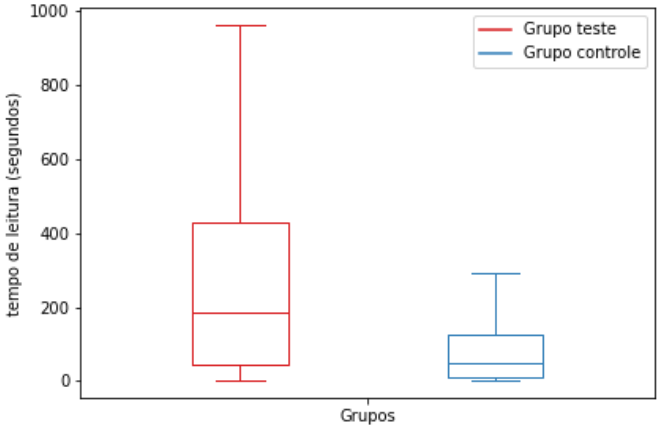

Figura 1. Comparação dos tempos de leitura dos grupos teste e controle

4.2. Q2: Há diferença estatisticamente significante na densidade de leitura do feedback entre os grupos?

O grupo controle recebeu apenas a tradução das CEMs, ao passo que o grupo teste recebeu, além da tradução, as ECEMs. Logo, um maior tempo de leitura neste último grupo pode ser explicado por haver mais palavras a serem lidas. Para verificar essa possibilidade, adotamos a métrica *densidade de leitura*, definida como a razão entre o tempo de leitura (em segundos) e o número de palavras contidas em cada mensagem de feedback.

Também foi encontrada diferença estatisticamente significante ($p < 0{,}05$) na densidade de leitura dos grupos teste e controle ($p < 0{,}0001$). E, como podemos observar na Figura 2, quando utilizamos a densidade de leitura proposta, o tempo do grupo teste fica menor do que o do grupo controle, podendo indicar que os estudantes que receberam apenas a tradução do erro passaram mais tempo tentando entendê-lo.

4.3. Q3: Há diferença estatisticamente significante na quantidade de visualizações do feedback entre os grupos?

Calculamos o número de aberturas de mensagens de feedback pelos grupos teste e controle, com os dados dispostos na Tabela 3. Obtivemos, na razão entre o total de

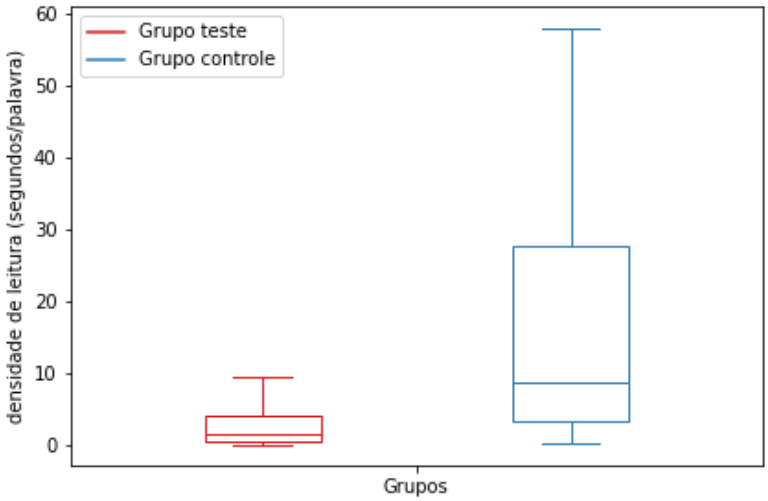

Figura 2. Comparação das densidades de leitura dos grupos teste e controle

visualizações e número de alunos do grupo, os valores de 22,95 visualizações para o grupo teste e 11,31 visualizações para o grupo controle. Percebemos que o grupo teste acessou mais vezes o conteúdo do feedback do que o grupo controle em quase todos os erros, exceto no erro E3, podendo indicar que o feedback criado resultou em uma correção mais rápida do erro para o aluno. Também obtivemos diferença estatística significante ($p < 0{,}05$) nas aberturas de feedback dos erros E2, E4, E7 e E12. Isso demonstra que o grupo teste acessou mais vezes o feedback do que o grupo controle nesses erros, o que pode indicar um interesse maior dos estudantes pelo conteúdo presente no feedback. No erro E9, não foi identificada nenhuma abertura de dica. Apesar dos resultados, em alguns erros, grande parte dos alunos dos dois grupos não visualizou os feedbacks, gerando boxplots com pouca variação.

Na visualização dos dados pelo boxplot, mostrada na Figura 3, percebe-se uma diferença estatisticamente significante na distribuição dos valores nos erros. Os dados apresentados foram padronizados para melhor visualização, por conta da diferença de ocorrência entre os tipos de erro.

4.4. Q4: Há diferença estatisticamente significante na quantidade de erros cometidos entre os grupos?

A partir do atributo nome_erro, calculamos o número de ocorrências de cada um dos erros pelos grupos teste e controle, com os dados dispostos na Tabela 4. Por meio da razão entre quantidade de erros e número de alunos de cada grupo, obtivemos os valores de 234,0 erros para o grupo teste e 202,45 para o grupo controle, podendo reforçar uma mesma conclusão apontada pelo artigo de referência: Os alunos do grupo teste testaram mais vezes o código.

Observamos que os erros E1, E4 e E12 apresentaram diferença estatisticamente significante ($p < 0{,}05$), indicando que a criação do feedback também pode ter influenciado os tipos de erros cometidos pelos alunos. Pela Figura 4, também observamos uma evidente diferença de distribuição nos erros com diferença estatística relevante. Os dados apresentados foram padronizados para melhor visualização, por conta da diferença de ocorrência entre os tipos de erro.

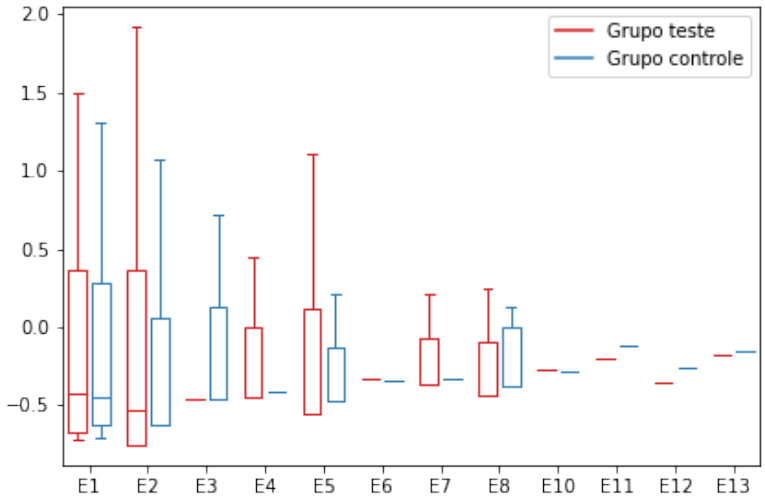

Figura 3. Comparação de visualizações do feedback pelos grupos teste e controle, divididas por tipo de erro

Tabela 3. Quantidades de aberturas de feedback, divididas por tipo de erro

#	Nome do erro	Grupo teste	Grupo controle	Valor p
E1	SyntaxError: invalid syntax	1070	623	0,06
E2	NameError: name <variable_name> is not defined	247	136	**0,01**
E3	IndentationError: unindent does not match any outer indentation level	46	58	0,27
E4	IndentationError: expected an indented block	73	40	**0,01**
E5	SyntaxError: unexpected EOF while parsing	121	103	0,22
E6	ValueError: invalid literal for int() with base 10	62	26	0,17
E7	IndentationError: unexpected indent	92	15	**0,01**
E8	TypeError: unsupported operand type(s)	94	55	0,21
E9	TypeError: Can't convert 'int' object to str implicitly	0	0	–
E10	TabError: inconsistent use of tabs and spaces in indentation	25	21	0,23
E11	ZeroDivisionError: division by zero	3	1	0,15
E12	IndexError: index X is out of bounds for axis Y with size Z	40	17	**0,04**
E13	ZeroDivisionError: float division by zero	9	3	0,31

4.5. Q5: Há diferença estatisticamente significante em como os alunos de cada grupo percebem a utilidade das mensagens de feedback?

Para complementar a pesquisa, mediu-se a satisfação dos alunos em relação ao feedback criado, tanto em conteúdo como em apresentação. Para responder a essa questão de pesquisa, criamos um questionário de opinião para os alunos que participaram do experimento. O questionário foi aplicado na metade do período letivo e era requisito para que o estudante continuasse realizando as atividades do curso. Dessa forma, apenas os alunos não desistentes responderam as questões. O questionário continha 8 afirmações. Para cada uma delas, os estudantes deveriam marcar sua concordância em uma escala Likert de 1 (discordo totalmente) a 5 (concordo totalmente). As afirmações foram dispostas aleatoriamente, de forma a minimizar algum viés relativo à ordem de apresentação. Elas

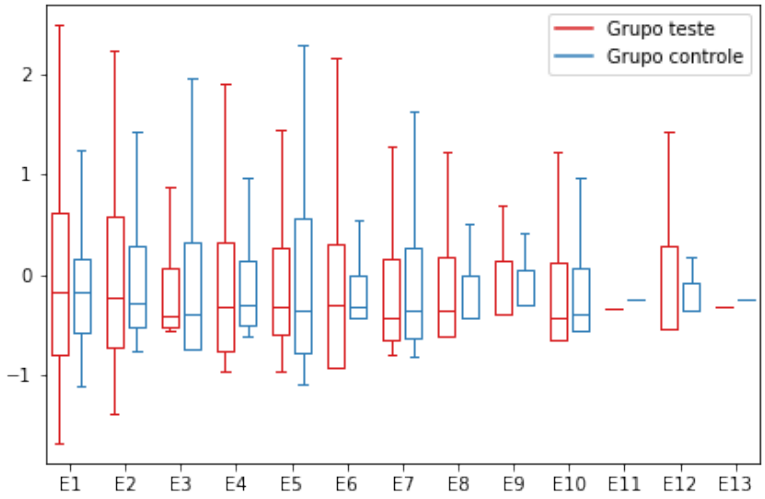

Figura 4. Comparação das quantidades de erros dos grupos teste e controle, divididas por tipo de erro

Tabela 4. Quantidades de erros de teste cometidos pelos alunos, divididos por tipo de erro

#	Nome do erro	Grupo teste	Grupo controle	Valor p
E1	SyntaxError: invalid syntax	12299	12443	**0,01**
E2	NameError: name <variable_name> is not defined	1363	1802	0,14
E3	IndentationError: unindent does not match any outer indentation level	942	795	0,34
E4	IndentationError: expected an indented block	1086	1035	**0,02**
E5	SyntaxError: unexpected EOF while parsing	1464	1270	0,05
E6	ValueError: invalid literal for int() with base 10	241	380	0,18
E7	IndentationError: unexpected indent	871	860	0,39
E8	TypeError: unsupported operand type(s)	189	219	0,17
E9	TypeError: Can't convert 'int' object to str implicitly	59	80	0,48
E10	TabError: inconsistent use of tabs and spaces in indentation	478	603	0,47
E11	ZeroDivisionError: division by zero	21	19	0,18
E12	IndexError: index X is out of bounds for axis Y with size Z	157	125	**0,01**
E13	ZeroDivisionError: float division by zero	18	7	0,11

abordavam a utilidade, apresentação visual e facilidade de entendimento do conteúdo:

A1: *O feedback é útil para resolver os exercícios.*
A2: *O feedback é útil para encontrar e resolver problemas de programação.*
A3: *O feedback é útil para aprender a programar.*
A4: *O texto do feedback é claro e objetivo.*
A5: *O texto do feedback é incompleto.*
A6: *O feedback é exibido na tela com facilidade de uso.*
A7: *O feedback se torna desnecessário após algum tempo usando-o.*
A8: *Eu tive problemas aplicando as recomendações sugeridas pelo feedback.*

Como mostrado na Tabela 5, não houve diferença significativa entre as respostas dos dois grupos. As médias e desvios dos estudantes de diferentes grupos foram similares

em cada uma das afirmações, indicando que mesmo que exista uma diferença entre os grupos nas outras métricas, isso foi avaliado de forma similar pelos estudantes. No entanto, existem afirmações que se posicionaram mais próximas das extremidades da escala, A1 tendo a pontuação mais alta e A7 a pontuação mais baixa, nos dois grupos. Isso pode nos indicar que o feedback foi considerado útil pelos estudantes e não foi considerado desnecessário ao longo do tempo, mesmo com a diferença de tratamento.

Tabela 5. Respostas dos alunos ao questionário de opinião

	Opinião sobre o feedback, por grupo	
Afirmação	**Grupo controle**	**Grupo teste**
A1	$3{,}93 \pm 1{,}02$	$3{,}98 \pm 1{,}12$
A2	$3{,}69 \pm 1{,}09$	$3{,}58 \pm 1{,}15$
A3	$3{,}79 \pm 1{,}14$	$3{,}75 \pm 1{,}09$
A4	$3{,}38 \pm 1{,}13$	$3{,}19 \pm 1{,}10$
A5	$2{,}94 \pm 1{,}13$	$3{,}02 \pm 1{,}08$
A6	$3{,}51 \pm 1{,}15$	$3{,}56 \pm 0{,}98$
A7	$2{,}59 \pm 1{,}36$	$2{,}41 \pm 1{,}24$
A8	$3{,}16 \pm 1{,}16$	$3{,}08 \pm 1{,}11$

4.6. Ameaças à Validade

Os problemas de instabilidade na rede da Universidade foram as principais ameaças à validade do experimento, pois em algumas ocasiões os dados de interação com as mensagens de feedback não foram devidamente salvas no servidor. Tivemos o cuidado de não incluir na análise os dados afetados por esses problemas, mas isso reduziu o número total de dados. Além disso, a alta taxa de desistência que historicamente vem se repetindo nas turmas de introdução à programação é um fator que limita o número de alunos que podem ser analisados, reduzindo a quantidade total de dados das amostras. Em adição a isso, apesar de termos estratificado os grupos apenas por sexo (masculino e feminino) para evitar um desbalanceamento, e distribuído os alunos entre as turmas que historicamente apresentam altas e baixas taxas de aprovação, temos variáveis sociais não consideradas no experimento (como dificuldade de leitura), que podem ter influenciado os resultados.

5. Conclusão e Trabalhos Futuros

No artigo de referência, identificamos que os estudantes que receberam o feedback estendido (grupo teste) testaram mais vezes o código, o que sugere maior interesse pelo conteúdo apresentado. No presente trabalho, observou-se que os estudantes também acessaram mais vezes o conteúdo e o leram por mais tempo do que os estudantes que receberam apenas a tradução das CEMs do Python (grupo controle). Além disso, também observamos diferença estatística significativa na ocorrência de alguns tipos de erro, sugerindo que o tratamento proposto pode ter interferência direta nos erros cometidos pelos estudantes que recebem o conteúdo. Apesar dos resultados, essa mudança parece não ter sido percebida pelos estudantes, como observamos na Tabela 5. Então, mesmo que haja mudança nos tipos de erros cometidos ou no aprendizado dos estudantes sobre o conteúdo em análise, isso não necessariamente vai ser observado por eles. Como trabalhos futuros, pretendemos ajustar o feedback com sugestões dos alunos e conforme novas evidências de efetividade em mensagens de feedback surgirem. Além disso, pretendemos realizar novas rodadas do experimento, incluindo mais turmas para análise.

6. Agradecimentos

Esta pesquisa, realizada no âmbito do Projeto Samsung-UFAM de Ensino e Pesquisa (SUPER), nos termos do artigo 48 do Decreto nº 6.008/2006 (SUFRAMA), foi parcialmente financiada pela Samsung Eletrônica da Amazônia Ltda., nos termos da Lei Federal nº 8.387/1991, por meio dos convênios 001/2020 e 003/2019, firmados com a Universidade Federal do Amazonas e a FAEPI, Brasil. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

Alves, F. P. and Jaques, P. (2014). Um ambiente virtual com feedback personalizado para apoio a disciplinas de programaçao. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 3, page 51.

Becker, B. A. (2016). An effective approach to enhancing compiler error messages. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 126–131.

Becker, B. A., Glanville, G., Iwashima, R., McDonnell, C., Goslin, K., and Mooney, C. (2016). Effective compiler error message enhancement for novice programming students. *Computer Science Education*, 26(2-3):148–175.

de Oliveira, J., Oliveira, E., de Carvalho, L. S. G., and Fernandes, D. (2019). Mensagens estendidas de feedback em um juiz online para alunos de introdução à computação: resultados preliminares. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 30, page 329.

Denny, P., Prather, J., and Becker, B. A. (2020). Error message readability and novice debugging performance. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, pages 480–486.

Jesus, G. S., Santos, K., Conceição, J., Ribeiro, E., and Neto, A. C. (2018). Avaliação de uma abordagem para auxiliar a correção de erros de aprendizes de programação. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, volume 29, pages 1–10.

Karvelas, I., Li, A., and Becker, B. A. (2020). The effects of compilation mechanisms and error message presentation on novice programmer behavior. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pages 759–765.

Keuning, H., Jeuring, J., and Heeren, B. (2018). A systematic literature review of automated feedback generation for programming exercises. *ACM Transactions on Computing Education (TOCE)*, 19(1):1–43.

Narciss, S. (2008). Feedback strategies for interactive learning tasks. *Handbook of research on educational communications and technology*, 3:125–144.

Nienaltowski, M.-H., Pedroni, M., and Meyer, B. (2008). Compiler error messages: What can help novices? In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, pages 168–172.

Prather, J., Pettit, R., McMurry, K. H., Peters, A., Homer, J., Simone, N., and Cohen, M. (2017). On novices' interaction with compiler error messages: A human factors approach. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*, pages 74–82.