Mensagens estendidas de feedback em um juiz online para alunos de introdução à computação: resultados preliminares

**Joseph V. L. de Oliveira, Elaine H. T. de Oliveira,
Leandro S. G. Carvalho, David B. F. Oliveira**

[1]Instituto de Computação – Universidade Federal do Amazonas (UFAM)
Av. General Rodrigo Octávio, 6200 – Coroado I, 69077-000 – Manaus, AM – Brasil

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

Resumo. *Este artigo objetiva verificar a efetividade de se apresentar mensagens de feedback estendido a alunos em um juiz online utilizado em disciplina de introdução à programação. Foi realizado um estudo de caso do tipo intervenção/controle, com 5 turmas de cursos de engenharia e ciências exatas, num total de 274 alunos matriculados, dos quais 179 permaneceram ativos. Não se observou diferença significativa entre alunos do grupo controle e experimental com respeito ao desempenho acadêmico nas quatro avaliações, nem ao tempo empregado pelos estudantes para resolver os exercícios práticos. Por outro lado, observou-se que o grupo que recebeu o feedback estendido realizou um maior número de testes de código que o grupo controle, indicando que o conteúdo foi mais visualizado.*

Abstract. *This article aims to verify the effectiveness of presenting extended feedback messages to students in an online judge used in programming introduction discipline. An intervention/control case study was carried out, with 5 classes of engineering and exact sciences courses, in a total of 274 students enrolled, of which 179 remained active. There was no significant difference between students in the control and experimental groups regarding the academic performance in the four evaluations, nor the time taken by the students to solve the practical exercises. On the other hand, it was observed that the group that received the extended feedback performed a greater number of code tests than the control group, indicating that the content was more visualized.*

1. Introdução

São cada vez maiores os índices de reprovação, desperiodização e de desistência em cursos de ciências exatas nas universidades brasileiras [Hoed 2016, Lobo 2017]. Por conta disso, o desempenho acadêmico dos alunos tem ganhado atenção crescente em áreas como análise de dados, mineração de dados e ciência de dados educacionais nos últimos anos. Essas áreas têm como objetivo processar dados para a obtenção de informações úteis, a fim de obter conclusões que irão dar suporte e sugerir tomadas de decisões.

Embora a finalidade das mensagens de erro seja ajudar o programador a localizar e corrigir erros, elas normalmente são inadequadas para os estudantes novatos [Denny et al. 2014]. Tais mensagens de erro, escritas em inglês, podem ser confusas ou vagas para muitos dos estudantes, além de necessitarem de um conhecimento mais técnico

para seu entendimento completo. Dessa forma, podem contribuir para sua frustração e até mesmo para abandono da disciplina [Roberto et al. 2018].

A presente pesquisa visa verificar, por meio de um estudo experimental do tipo intervenção/controle, se a apresentação de mensagens de feedback estendido (conteúdo criado, que substitui a mensagem original do compilador) sobre erros de programação da linguagem Python apresenta impacto estatisticamente significativo sobre indicadores de desempenho de estudantes dos cursos de IPC (Introdução à Programação de Computadores) em ambiente de correção automática de códigos. Para isso, foram analisados os dados de submissão dos alunos da plataforma "Codebench", detalhados com mais profundidade na seção de metodologia. Para guiar a análise dos experimentos, levantamos as seguintes questões de pesquisa:

- **Q1:** Há diferença no número de testes e de submissões entre os alunos de cada grupo (experimental e controle)?
- **Q2:** Há diferença de desempenho acadêmico (nota dos trabalhos práticos) entre os alunos de cada grupo (experimental e controle)?
- **Q3:** Há diferença no tempo de resolução entre os alunos de cada grupo (experimental e controle)?

Como estudo de caso, foi utilizado o Codebench, um ambiente de correção automática de códigos, também conhecido como juiz online. Ele foi desenvolvido pelo Instituto de Computação - IComp, da Universidade Federal do Amazonas, com o objetivo de corrigir atividades submetidas por alunos alunos de disciplinas de programação. Atualmente, o Codebench tem cerca de 3.571 usuários cadastrados, 179 turmas criadas desde 2015, 3.140 exercícios disponíveis para os professores e 3 instituições parceiras[1].

O restante deste artigo está organizado da seguinte forma: Na Seção 2, é apresentada uma breve revisão da literatura sobre geração de mensagens para erros de programação. Na Seção 3, descreve-se os procedimentos metodológicos que estruturam esta pesquisa. Na Seção 4, apresentam-se os resultados preliminares, anteriores à conclusão das turmas estudadas. Por fim, na Seção 5, são apresentadas as conclusões e trabalhos futuros.

2. Revisão da Literatura

O grande diferencial entre o presente trabalho e os anteriores reside no método de tradução do código de programação em código executável pela máquina. [Becker 2016] utilizou um compilador para a linguagem de programação Java, e [Pettit et al. 2017] utilizaram um compilador para a linguagem C++. Na Universidade Federal do Amazonas, os cursos de IPC ministrados utilizam a linguagem de programação Python, cujos códigos são executados por meio de um interpretador, e não um compilador. Há também o trabalho realizado por [de Jesus et al. 2018], que utilizou a linguagem Python; porém, o trabalho conta com a utilização de uma amostra de apenas 26 participantes e somente com os alunos presentes no momento, não com uma amostra censitária. Além disso, a análise levou em consideração apenas a resolução de 6 exercícios (3 para cada grupo).

Classicamente, as linguagens de programação são classificadas em dois grupos, de acordo com o método que utilizam para traduzir o código de programação em código

[1]Codebench `http://codebench.icomp.ufam.edu.br`.

executável pela máquina [Sebesta 2018]: linguagens compiladas e linguagens interpretadas. Compilador é um programa que traduz código-fonte escrito em uma linguagem de programação de alto nível (semelhante a uma linguagem natural utilizada por seres humanos) para código objeto, que é um código binário executado pelo computador. Por outro lado, o interpretador não converte todo o código de alto nível para código objeto de uma vez. Ele executa diretamente cada instrução, passo a passo. Dessa forma, compiladores são capazes de encontrar vários erros de sintaxe de uma vez só, já que analisa todo o código de alto nível. Esse fato é colocado na literatura como uma ameaça à validade dos experimentos com mensagens de feedback aprimorado, já que eles explicam um erro de cada vez e as mensagens de erro dos compiladores podem ser mais úteis para alguns alunos por mostrarem mais de um erro simultaneamente [Denny et al. 2014].

Outro diferencial do presente trabalho em relação a de [Pettit et al. 2017] é que o interpretador do Codebench é utilizado como ferramenta padrão para os alunos desenvolverem códigos, ao passo que a ferramenta de [Pettit et al. 2017] é utilizada mais para correção dos códigos normalmente elaborados em ambientes que não possuem a funcionalidade de feedback estendido.

A maioria do material relevante encontrado na literatura estuda e realiza experimentos em linguagens como C e Java, porém possuem possível capacidade de reprodutibilidade para experimentos em outras linguagens de programação. Além disso, o material proposto na literatura não conseguiu alcançar, de forma conclusiva, um formato de feedback estendido que seja efetivo nos aspectos delineados nas questões de pesquisa. Sendo assim, esses são os elementos diferenciais propostos neste artigo.

3. Metodologia

Esta seção relata a metodologia utilizada para medir a efetividade de se exibir mensagens de feedback estendido para alunos das turmas de IPC em um ambiente de correção automática de código na linguagem Python.

3.1. Contexto

Para testar a efetividade das dicas sobre o desempenho dos alunos, utilizamos o juiz online empregado na instituição para prática de exercícios e avaliações na disciplina IPC, durante o primeiro período letivo de 2019. Essa disciplina foi escolhida porque é oferecida anualmente para 17 turmas de graduação em ciências exatas e engenharia, 11 no primeiro e 6 no segundo semestre letivo.

O juiz online sofreu duas grandes mudanças entre o ano de 2018 e 2019: além da introdução do feedback estendido, teve seu sistema de gamificação totalmente remodelado. Dessa forma, o experimento não pôde seguir o desenho de comparação de resultados entre anos diferentes, pois havia pelo menos duas variáveis de observação envolvidas.

Para contornar essa ameaça à validade, foi adotado um desenho experimental com grupo controle. Os estudantes matriculados em 5 das 11 turmas de IPC em 2019/1 foram divididos em dois grupos de tamanhos semelhantes: experimental e controle. As 6 turmas restantes foram alocadas para a análise da gamificação do Codebench.

À medida que os estudantes dessas 5 turmas se matriculavam no juiz online, eles eram alternadamente designados para o grupo controle ou o grupo experimental. Esses

grupos foram estratificados apenas por sexo (masculino e feminino), para evitar um desbalanceamento. A distribuição de turmas entre os dois estudos procurou mesclar turmas que historicamente apresentam altas e baixas taxas de aprovação.

Para os estudantes do grupo **experimental**, o juiz online exibe dois tipos de informação quando o interpretador Python identifica um dos 14 erros selecionados (Tabela 1): as **mensagens de erro do interpretador traduzidas para o português** e **mensagens de feedback estendido**. Para os estudantes do grupo **controle**, o juiz online exibe apenas as **mensagens traduzidas**, sem explicações adicionais. Essa configuração foi mantida nos 4 primeiros módulos da disciplina, equivalente à primeira metade do período letivo. Nos 3 últimos módulos, o juiz online foi programado para inverter a regra de exibição de mensagens de feedback. Isso garante a integridade ética da pesquisa, já que todos os estudantes terão acesso ao mesmo conteúdo, ainda que em tempos diferentes. Além disso, na plataforma online, há a opção de "teste" e "submissão" dos exercícios, para podermos realizar uma análise mais detalhada. O conteúdo de feedback estendido e de tradução só é mostrado ao usuário pela opção de "teste", a "submissão" tem o propósito de submeter o exercício previamente testado.

3.2. Coleta e análise dos erros dos alunos

O primeiro passo para a elaboração das mensagens de feedback estendido foi identificar os erros mais frequentes cometidos pelos alunos na disciplina de IPC, na linguagem Python, durante sua interação com o juiz online, de 2016/1 a 2018/1 (cinco períodos letivos). Verificou-se a ocorrência de 129.780 submissões com erros, distribuídos entre mais de 90 tipos de erro de codificação em Python.

Tabela 1. Erros de codificação mais comuns cometidos pelos alunos de IPC entre 2016/1 e 2018/1 durante a submissão de códigos Python ao juiz online

#	Nome do erro	Ocorrências	%
1	SyntaxError: invalid syntax	32.342	24,92%
2	NameError: name <nome_da_variável> is not defined	27.334	21,06%
3	ValueError: invalid literal for int() with base 10	9.814	7,56%
4	EOFError: EOF when reading a line	7.923	6,10%
5	IndentationError: expected an indented block	4.254	3,28%
6	IndentationError: unindent does not match any outer indentation level	4.080	3,14%
7	IndentationError: unexpected indent	3.676	2,83%
8	SyntaxError: unexpected EOF while parsing	3.580	2,76%
9	IndexError: index X is out of bounds for axis Y with size Z	2.903	2,24%
10	TabError: inconsistent use of tabs and spaces in indentation	2.598	2,00%
11	ZeroDivisionError: float division by zero	2.349	1,81%
12	TypeError: unsupported operand type(s)	2.085	1,61%
13	ZeroDivisionError: division by zero	244	0,19%
14	TypeError: Can't convert 'int' object to str implicitly	144	0,11%

A princípio, definimos um número mínimo de 2.000 ocorrências como critério para a elaborar mensagem de feedback estendido, a fim de concentrar esforços nos erros mais relevantes e frequentes. As únicas exceções foram os erros ZeroDivisionError: division by zero e TypeError: Can't convert 'int' object to str implicitly, pois foram de fácil confecção, devido à existência de erros similares entre os erros descritos. Dessa forma, foram elaboradas mensagens de feedback estendido para os 14 erros listados na

Tabela 1, que correspondem a 79,61% do número total de erros cometidos de 2016/1 a 2018/1.

3.3. Criação das mensagens de feedback estendido

O conteúdo do feedback estendido foi elaborado seguindo os princípios de geração de feedback descritos por [Shute 2008], debatidos com mais profundidade em pesquisas relacionadas à área de psicologia. A estrutura e a apresentação do conteúdo foi parcialmente baseada em pesquisas realizadas por [Becker 2016] com a ferramenta DECAF, bem como nas limitações de visualização do Codebench.

Após esse passo, o conteúdo foi revisado por programadores mais experientes na linguagem Python, por meio de um questionário qualitativo. Esse questionário foi respondido por 24 pessoas, com diferentes experiências e perspectivas sobre o problema (Figura 1). Em seguida, o conteúdo das mensagens de feedback foi modificado seguindo as sugestões mais pertinentes e com o que foi observado ser semelhante ao proposto na literatura científica relacionada.

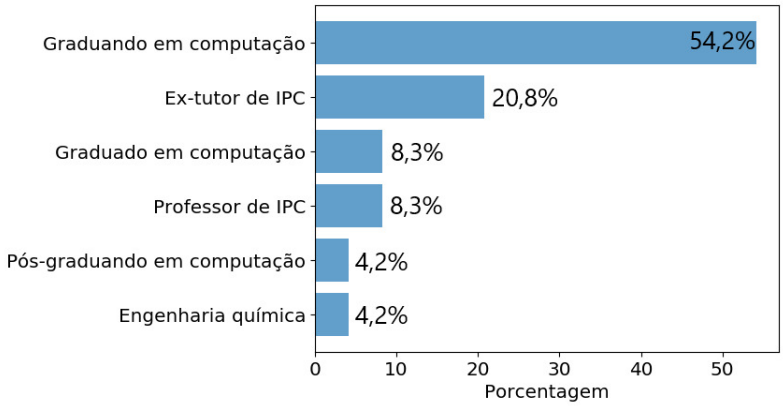

Figura 1. Tipos de especialistas que responderam o questionário.

3.4. Estrutura do feedback estendido

Cada mensagem de feedback estendido adota a seguinte estrutura, conforme pode ser visto na Figura 2:

1. Título (mensagem de erro gerada pelo interpretador Python, em inglês);
2. Tradução para o português;
3. Breve descrição sobre a possível causa do erro e uma possível solução; e
4. Dois códigos de exemplo: Um código contendo o erro e um código com uma solução para o erro detectado.

3.5. Desenho do experimento

Como o experimento envolveu a participação de seres humanos, o projeto foi submetido ao Comitê de Ética em Pesquisa por meio da Plataforma Brasil sob o número CAAE 08810619.1.0000.5020. Após a aprovação, aguardou-se o início do período letivo de 2019/1 para interação com os alunos, que foram informados sobre a pesquisa e convidados a participar, assinando um Termo de Consentimento Livre e Esclarecido. Não foi observada nenhuma objeção.

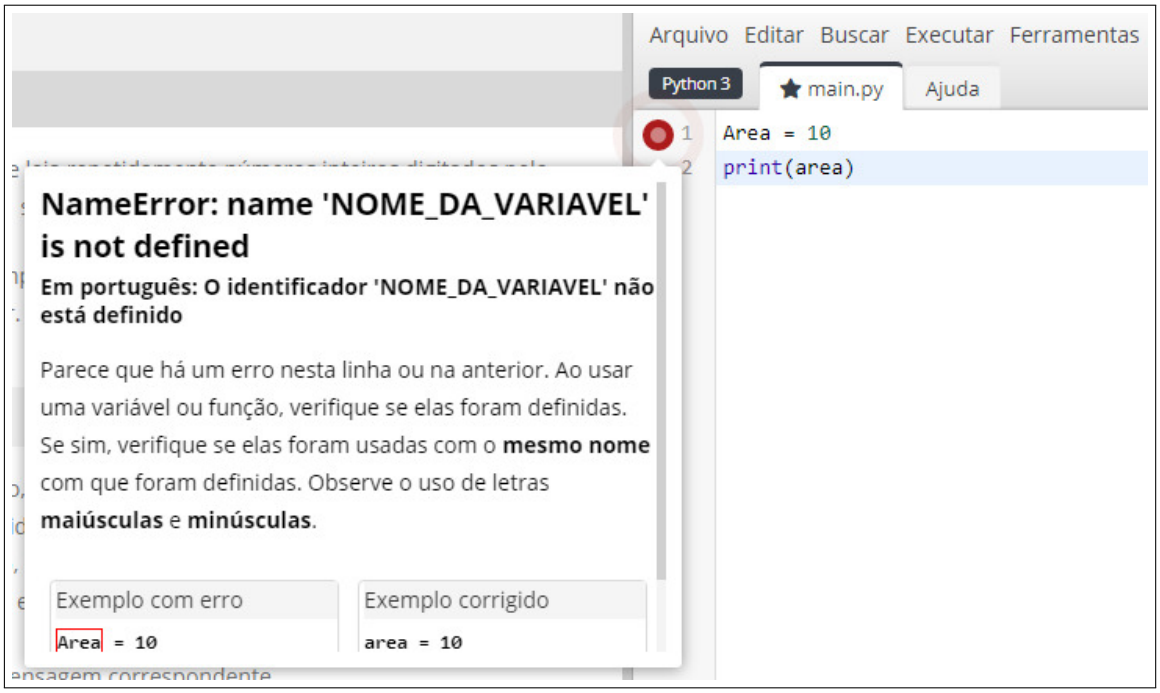

Figura 2. Exemplo de apresentação de um erro com feedback estendido.

A disciplina é dividida em 7 módulos, compostos por uma aula teórica, duas práticas e uma avaliação. Os exercícios práticos têm natureza formativa e valem 0,9% da nota final dos estudantes. As avaliações, de natureza somativa, pesam de 4% a 12% da nota final, à medida que o grau de dificuldade do conteúdo aumenta.

O experimento foi realizado coletando-se os dados de interação dos alunos com o ambiente de correção automática de códigos durante a realização dos exercícios práticos, e os dados de desempenho acadêmico (nota) nas avaliações de cada módulo. No caso dos exercícios práticos, que podiam ser resolvidos fora da sala de aula, foram excluídos os dados de interação gerados por alunos que apresentavam poucas linhas de log, indicativo de apenas copiaram e colaram código de outro colega. No caso das avaliações, foram considerados todos os alunos ativos até o final do módulo 4 (20 de maio de 2019), data de corte para a realização do presente estudo. É importante salientar que o feedback estendido só é mostrado nos exercícios práticos, e não nas avaliações. A Tabela 2 apresenta os dados coletados a partir do juiz online.

4. Resultados

Nesta seção, são apresentados os resultados iniciais da pesquisa, com base nos dados coletados até o TP4 entre os sete Trabalhos Práticos. Eles estão estruturados conforme as três perguntas de pesquisa apresentadas na Introdução.

4.1. Caracterização da Amostra

O estudo foi iniciado com 274 alunos, divididos entre as 5 turmas participantes. Destes, verificou-se que 95 alunos eram desistentes, e seus dados foram removidos da análise, restando 179 alunos que continuaram participando da disciplina. Entre as 5 turmas, 4 são de calouros, de modo que historicamente se observa uma elevada desistência do curso

Tabela 2. Atributos observados neste estudo, coletados a partir do juiz online

Atributo	Significado
tipo_dica	*flag* que indica se o aluno recebeu feedback estendido ou tradução
sexo	sexo do aluno
nascimento	ano de nascimento do aluno
respondeu_questionario	*flag* que indica se o aluno respondeu o questionário obrigatório após o Trabalho Prático 4. Foi utilizada para filtrar os alunos desistentes
tp_1	nota no Trabalho Prático 1 (prova 1)
tp_2	nota no Trabalho Prático 2 (prova 2)
tp_3	nota no Trabalho Prático 3 (prova 3)
tp_4	nota no Trabalho Prático 4 (prova 4)
n_testes	número de testes de código feitos pelo aluno no juiz online
n_submissoes	número de submissões de código feitas pelo aluno no juiz online
n_exercicios_corretos	número de exercícios corretos do aluno
total_tempo_ide	tempo total gasto para resolver os exercícios corretos
media_tp	média das notas dos trabalhos práticos 1,2,3 e 4

e, consequentemente, da disciplina. Os 179 alunos restantes estão caracterizados como mostra a Tabela 3.

Tabela 3. Caracterização dos alunos participantes do experimento

	Característica			
Grupo	**Homens**	**Mulheres**	**Total**	**Idade (anos)**
Experimental	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	$20{,}4 \pm 4{,}21$

4.2. Comparação de Desempenho

Efetuou-se um teste de Shapiro-Wilk para verificar a normalidade dos dados. Constatou-se que nenhum dos parâmetros seguiam uma distribuição normal.

Em seguida, aplicou-se o teste não paramétrico de Mann-Whitney (não pareado) para verificar se havia diferença significativa entre os grupos experimental (feedback estendido) e controle (tradução). Notou-se que houve diferença significativa apenas com respeito ao número de testes de código, não se verificando diferença significativa com respeito aos demais fatores:

- Nota dos trabalhos práticos;
- Tempo de resolução;
- Número de submissões.

4.3. Q1: Há diferença no número de testes e de submissões entre os alunos de cada grupo (experimental e controle)?

Como pudemos observar nos resultados obtidos pela aplicação do teste não paramétrico de Mann-Whitney (não pareado), na Tabela 4, o número de testes obteve um valor p de 0,04, indicando que houve diferença significativa nesse atributo entre os grupos. Observa-se, na Figura 3, que a quantidade de testes feitos pelo grupo teste é superior à do grupo

Tabela 4. Valores-p resultantes do teste de Mann-Whitney entre parâmetros dos grupos controle (tradução) e experimental (dica completa)

Atributo	valor *p*
Nota do TP 1	0,20
Nota do TP 2	0,96
Nota do TP 3	0,19
Nota do TP 4	0,07
Média das notas dos TPs	0,18
N° de testes	**0,04**
N° de submissões	0,42
N° de exercícios corretos	0,22
Tempo total de IDE	0,10

controle, comportamento que não é percebido com tanta significância na quantidade de submissões, e que não foi apontado pelo teste não paramétrico como um atributo relevante. Isso indica que as mensagens de feedback estendido foram visualizadas mais vezes do que as mensagens de tradução, podendo indicar um interesse maior dos alunos pelo conteúdo adicional do feedback estendido.

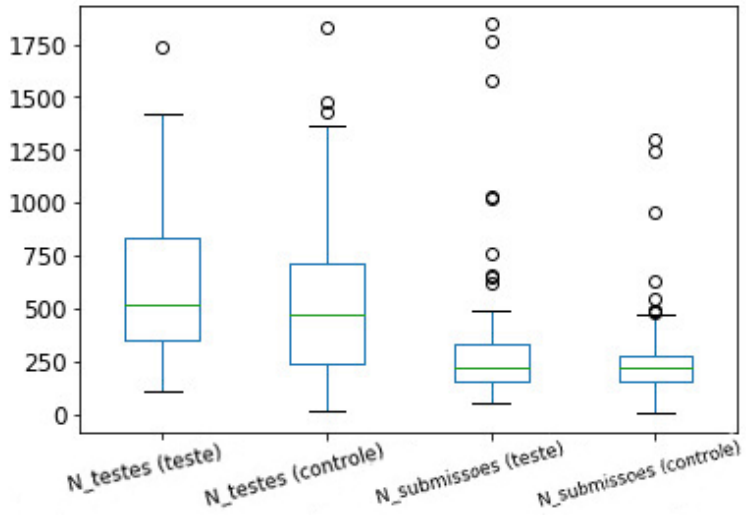

Figura 3. Distribuição do número médio de testes e de submissões por aluno participante dos grupos controle e experimental, realizados até o módulo 4.

4.4. Q2: Há diferença de desempenho acadêmico (nota dos trabalhos práticos) entre os alunos de cada grupo (experimental e controle)?

Comparando dados dos TPs 1 ao 4, obtidos a partir da interação dos alunos com o juiz online e dispostos na Tabela 5, observa-se que os alunos que receberam o feedback estendido tiraram, em média, 0,72 pontos a mais que os alunos que não receberam o conteúdo. No entanto, o valor observado não foi muito relevante na análise. Além disso, a variável não foi dada como relevante no teste não paramétrico nem segue uma distribuição normal.

Tabela 5. Evolução das notas nas quatro avaliações realizadas, entre as sete previstas, no grupo controle e experimental

	Grupo	
Avaliação	**Controle**	**Experimental**
TP1	5,97 ± 4,60	6,89 ± 4,20
TP2	6,39 ± 4,43	6,52 ± 3,98
TP3	4,58 ± 4,65	5,48 ± 4,34
TP4	2,70 ± 3,91	3,62 ± 4,04
Média	4,91 ± 0,33	5,63 ± 0,16

4.5. Q3: Há diferença no tempo de resolução entre os alunos de cada grupo (experimental e controle)?

A Figura 4 mostra que não foi observada uma diferença relevante no tempo de resolução entre os alunos de cada grupo e não houve uma presença relevante de outliers. Nosso pensamento inicial era de que, com a quantidade maior de informação do feedback estendido, o grupo teste teria um tempo significativamente menor de resolução dos exercícios, mas isso não foi percebido na análise.

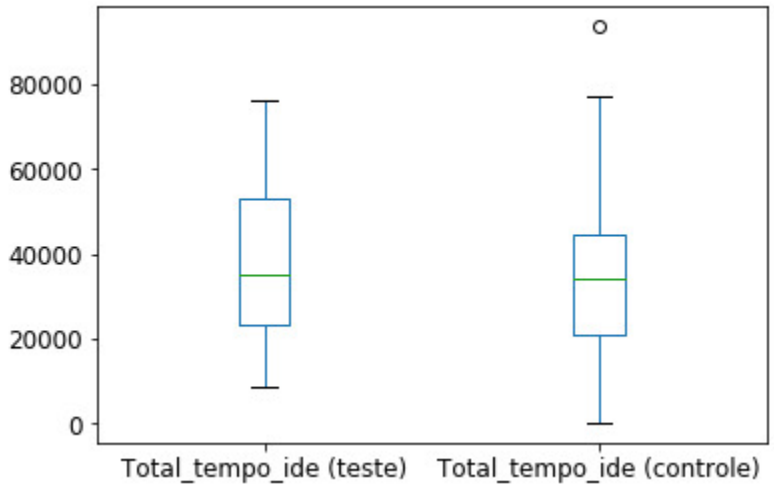

Figura 4. Tempo total utilizado na IDE, em segundos (somente dos exercícios corretos).

4.6. Ameaças à Validade

Como todo experimento com a participação de seres humanos, existem variáveis relacionadas aos participantes que estão fora de nosso controle, por exemplo: condição socioeconômica dos alunos, presença nas aulas, conhecimentos adquiridos via educação básica, controle efetivo de plágio nos exercícios, etc. São variáveis sociais que não foram consideradas nos dados analisados e que podem alterar o desempenho e interação do aluno com a plataforma utilizada no curso e, portanto, alterando o resultado final da análise.

Neste experimento, foi necessário adotar o modelo de intervenção/controle para tentar minimizar ao máximo as ameaças à validade que estavam, até certo ponto, sob

nosso controle. No entanto, a quantidade limitada de estudantes disponíveis para o experimento (o que gerou uma quantidade limitada de dados) também pode ter comprometido a representatividade dos resultados finais. Além disso, como se trata de uma análise parcial, os resultados com o conjunto completo de dados podem gerar resultados diferentes.

5. Conclusão e Trabalhos Futuros

Foi observado que os alunos que receberam o feedback estendido testaram mais vezes o seu código que os alunos que receberam somente a tradução, podendo indicar uma preferência pelo conteúdo contido nas mensagens de feedback estendido, mesmo não tendo sido identificada uma melhora significativa nas notas dos trabalhos práticos. Porém, esses são os resultados de uma análise parcial, sendo necessário esperar o fim do período letivo de 2019/1 para ter a totalidade dos dados e aplicar novamente os testes estatísticos. Como próximos passos, faremos a análise de um questionário, que foi respondido por todos os alunos participantes do experimento, para obter dados qualitativos sobre o feedback estendido, além da análise completa dos dados após o fim do período letivo. Além disso, faremos a análise comparativa dos conjuntos de erros (quais erros foram mais cometidos por cada um dos grupos).

Agradecimentos

Os autores agradecem ao apoio prestado pela Fundação de Amparo à Pesquisa do Estado do Amazonas - FAPEAM por meio do Edital N. 002/2018 - Universal Amazonas.

Referências

Becker, B. A. (2016). An Effective Approach to Enhancing Compiler Error Messages. *Proceedings of the 47th ACM SIGCSE '16*, pages 126–131.

de 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, page 1.

Denny, P., Luxton-Reilly, A., and Carpenter, D. (2014). Enhancing syntax error messages appears ineffectual. In *Proceedings of the 2014 conference on Innovation & technology in computer science education*, pages 273–278. ACM.

Hoed, R. M. (2016). Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de computação. *Brasília, DF: Universidade de Brasília*.

Lobo, R. (2017). A evasão no ensino superior brasileiro – novos dados. estadão.edu, 07/10/2017. disponível em: http://educacao.estadao.com.br/blogs/roberto-lobo/497-2/.

Pettit, R. S., Homer, J., and Gee, R. (2017). Do enhanced compiler error messages help students?: Results inconclusive. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, pages 465–470. ACM.

Roberto, G. F., Oliveira, F. S., Pinto, P. E. D., and Coelho, I. M. (2018). Tupy online-programação em português com visualização de execução e abstraçõ es de estruturas de dados na web. In *26º Workshop sobre Educação em Computação (WEI 2018)*. SBC.

Sebesta, R. W. (2018). *Conceitos de Linguagens de Programação*. Bookman, 11 edition.

Shute, V. J. (2008). Focus on formative feedback. *Review of educational research*, 78(1):153–189.