

Maior Frequência na Aplicação de Instrumentos de Avaliação em uma Disciplina Introdutória de Programação: Impactos no Desempenho e Motivação

Adilson Vahldick¹, Pablo Schoeffel^{1,2}, Fernando Borguesan Liz¹, Vinicius Faria Culmant Ramos³, Raul Sidnei Wazlawick²

¹ Departamento de Engenharia de Software
Universidade do Estado de Santa Catarina (UDESC) – Ibirama, SC – Brasil

² Programa de Pós-Graduação em Ciência da Computação
Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC – Brasil

³ Departamento de Tecnologias da Informação e Comunicação
Universidade Federal de Santa Catarina (UFSC) – Araranguá, SC – Brasil

{adilson.vahldick, pablo.schoeffel}@udesc.br,
fernando.borguesan.liz@gmail.com, raul@inf.ufsc.br, v.ramos@ufsc.br

Abstract. *The individualized support by the teacher is unfeasible with classes around 40 students in the introductory programming courses. This paper reports an attempt to increase the frequency of assessments with an Introductory Programming class by creating smaller intervals between evaluations so that both students and teachers were aware of the evolution of the class throughout the semester. The article presents the results regarding students' performance and motivation, identifying a correlation between the motivation and the performance of the evaluations, but identifying unsatisfactory results in relation to the final averages and approval rates.*

Resumo. *O atendimento individualizado pelo professor é inviável com turmas em torno de 40 alunos nas disciplinas introdutórias de programação. Esse artigo relata uma tentativa de aumentar a frequência de avaliações com uma turma de Introdução à Programação, criando intervalos menores entre os instrumentos de avaliação (provas), para que tanto os alunos quanto professor estivessem cientes da evolução da turma durante todo o semestre. O artigo apresenta os resultados quanto ao desempenho e motivação dos alunos, identificando uma correlação entre a motivação e o desempenho das avaliações, porém identificando resultados não satisfatórios com relação às médias finais e taxas de aprovação.*

1. Introdução

O ensino de programação objetiva nos alunos o desenvolvimento de competências necessárias para conceber programas que resolvam problemas reais (Gomes et al., 2015). Existem três tipos essenciais de competências a serem apropriadas: compreensão de programas, a escrita e a depuração (Robins et al., 2003; Pears et al., 2007). Adquirir essas competências e habilidades exige muita disciplina e persistência do aluno e ao mesmo tempo atenção e acompanhamento por parte dos professores. Na tentativa de atenuar esse

esforço e vencer as dificuldades tanto por parte dos alunos quanto dos professores, existem inúmeros trabalhos propondo ferramentas e estratégias para auxiliar no processo de ensino e aprendizagem, como relatado em (Cristina & Aureliano, 2012).

Porém, na maioria das vezes os professores continuam usando estratégias didáticas habituais, como as aulas expositivas, resolução de listas de exercícios, provas teóricas e práticas. Uma das dificuldades nessas estratégias decorrem do grande número de estudantes sob a supervisão dos professores que acompanham as atividades dos alunos (Menezes et al., 2002; Gomes & Mendes, 2007). Em consequência, pela quantidade de trabalho necessário com a correção e avaliação, normalmente o professor acaba por limitar a quantidade de avaliações. Alguns trabalhos propõem alternativas para o problema de turmas grandes: combinação de estratégias pedagógicas com a estrutura do curso e das tarefas de aprendizagem em ambientes MOOC (Vihavainen et al., 2012); uso de uma ferramenta¹ web atualmente adotada por várias instituições para automatizar a correção, feedback e classificação dos alunos (Paes et al., 2013); e alunos sendo avaliadores das soluções dos seus colegas (Moreira, 2014).

Em contrapartida a esses trabalhos correlatos, este artigo apresenta o relato da experiência em uma disciplina com 47 alunos, em que foram aplicadas 8 provas durante 18 semanas de aula, com a intenção de manter os alunos motivados em estudar programação. Essa alta frequência de avaliações proporcionou uma granularidade mais fina quanto aos assuntos tratados em cada prova, ou seja, menos assuntos novos são abordados em cada avaliação, ao mesmo tempo que exigiu maior esforço por parte dos alunos em manterem-se estudando. Também demandou mais trabalho por parte do professor durante o semestre em aplicar e corrigir essas avaliações. O objetivo do trabalho foi avaliar a melhora no desempenho e motivação dos alunos com uma maior frequência de avaliações. Os resultados podem apontar a real necessidade da avaliação constante, uma vez que é preciso mantê-los motivados para persistirem em seus estudos.

O artigo encontra-se estruturado da seguinte forma: nessa primeira seção foi contextualizado o problema, apresentados alguns trabalhos correlatos, e descrito o objetivo do trabalho; na Seção 2 é apresentada a metodologia usada; na Seção 3 os resultados são apresentados e discutidos; as conclusões são apresentadas na Seção 4.

2. Método

Este trabalho descreve um estudo de caso realizado no primeiro semestre de 2018 em uma disciplina de Introdução à Programação no curso de Bacharelado em Engenharia de Software na Universidade do Estado de Santa Catarina oferecida no período noturno. A disciplina tem carga horária de 108 horas/aula com 3 encontros semanais de 2 horas/aula, totalizando 54 aulas distribuídas em 18 semanas. O primeiro encontro semanal acontecia em sala de aula usado para apresentação de novos conteúdos e/ou correções de exercícios. O segundo e terceiro encontros semanais ocorriam em laboratório, cujas aulas eram destinadas à resolução dos exercícios. Frequentaram na disciplina 47 alunos (43 homens e 4 mulheres), dos quais 29,17% (n=14) eram repetentes, pois já haviam cursado a disciplina anteriormente. Um aluno monitor estava disponível para dar suporte aos alunos em horário extraclasse.

¹ <https://www.thehuxley.com/>

Com relação às estratégias didáticas, a disciplina foi dividida em três momentos, como será detalhado a seguir e esquematizado na Figura 1, onde cada circunferência representa 1 semana ou 3 encontros, e as legendas internas representam as avaliações realizadas.

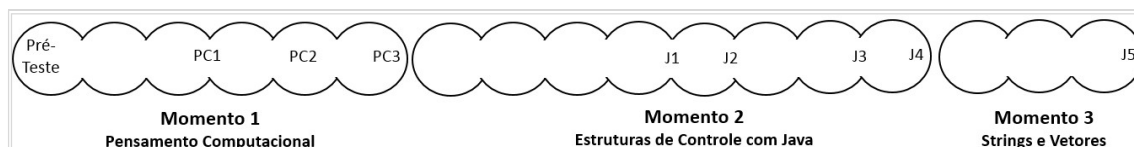


Figura 1. Distribuição das aulas e avaliações

Momento 1. Correspondeu aos 18 primeiros encontros, os quais foram dedicados ao aprendizado das habilidades de Pensamento Computacional usando o App Inventor², as atividades livres do Code Studio³, e o jogo sério NoBug's Snack Bar⁴ (Vahldick et al., 2017). Nas aulas teóricas foram resolvidos desafios e quebra-cabeças, ao mesmo tempo que as estruturas de controle iam sendo apresentadas através da sintaxe em blocos. Os exercícios em laboratório foram resolvidos com Code Studio e App Inventor. O jogo NoBug's Snack Bar é uma aplicação web e foi hospedada no servidor da Universidade. Os alunos podiam acessá-lo em qualquer período de qualquer local. Por essa razão, resolver os problemas dentro do jogo acabou sendo uma atividade extraclasse. O jogo estava dividido em 10 fases: 1-ambientação do jogo, 2 a 4-sequenciamento e manipulação de variáveis, 5 a 7-condicionais e 8 a 10-ciclos. No momento 1 foram aplicadas três avaliações: as duas primeiras em laboratório (PC1 nas aulas 08 e 09 e PC2 nas aulas 14 e 15) e a última resolvida em papel (PC3 na aula 18). As duas primeiras avaliações foram missões a serem realizadas no jogo, especialmente elaboradas para serem aplicadas como provas. A avaliação PC1 contemplava os assuntos tratados nas fases 2 a 4 do jogo e a avaliação PC2 contemplou os assuntos tratados nas fases 5 a 7. Como o laboratório não comportava os 47 alunos simultaneamente em computadores individualizados, a turma foi dividida em dois grupos. A PC3 foi uma avaliação em papel.

Momento 2. Correspondeu às 26 aulas dedicadas à manipulação de variáveis, estruturas de controle e métodos em linguagem de programação Java utilizando o framework Furbot⁵ (Vahldick & Mattos, 2009). Foram aplicadas 4 avaliações: duas no papel (J1 e J3) e duas usando computador (J2 e J4). As avaliações J1 (na aula 28) e J3 (na aula 39) tinham duas questões: a primeira sobre interpretação de código, onde os alunos deviam apontar as células percorridas pelo robô e a segunda, na qual os alunos deveriam construir um código em Java. As avaliações J2 (aulas 31 e 32) e J4 (aulas 43 e 44) tinham uma única questão para construir um programa em Java. Nestas avaliações em laboratório, a turma foi dividida em dois grupos em razão da sua capacidade não comportar a turma inteira: um grupo fazia numa aula, e o outro grupo na outra aula.

Momento 3. Correspondeu às 10 aulas dedicadas ao assunto final da ementa, ainda usando Java, mas sem nenhum framework adicional, com a utilização dos comandos básicos de entrada e saída, manipulação de caracteres (String), vetores e matrizes. Foi

² <http://appinventor.mit.edu/>

³ <https://www.code.org>

⁴ <https://github.com/adilsonv77/nobugssnackbar>

⁵ <http://www.inf.furb.br/poo/furbot/>

aplicada uma última avaliação (J5) com duas questões, em que os alunos tinham de implementar uma classe e um método para cada questão.

Houve 8 avaliações no total, o que garantiu uma granularidade fina quanto ao assunto aplicado em cada avaliação, exigindo um trabalho extra ao professor para sua elaboração e correção. Para verificar o impacto das avaliações na manutenção da motivação durante o semestre, foram formuladas três questões de análise:

Q1- A participação nas missões do jogo tem relação como desempenho nas avaliações?

Q2- A motivação na disciplina está relacionada ao resultado das avaliações?

Q3- O aumento no número de avaliações melhorou o aproveitamento dos estudantes?

Para responder a primeira questão, foi avaliado o número de missões realizadas correspondentes a cada avaliação (PC1, PC2 e PC3) e a sua correlação com o desempenho nessas mesmas avaliações.

Para as questão 2 foi elaborado um questionário de Avaliação Semanal da Motivação do Estudante⁶, contendo seis perguntas dividida em três grupos (Tabela 1), adaptado da Escala EVC (Kosovich et al., 2015). Eccles et al. (1983) definem que a dimensão de expectativa inclui a crença da habilidade (o que os estudantes acham que podem fazer agora) e crença na expectativa (o que os estudantes acham que conseguirão fazer no futuro). Já a dimensão do valor é definida pela importância da atividade em ajudar a atingir objetivos futuros e a dimensão de custo reflete um aspecto negativo de envolvimento na atividade (percepção de tempo e esforço requerido). Foi então calculada a correlação entre o índice de motivação e o desempenho nas avaliações, utilizando o coeficiente de correlação de Pearson.

Tabela 1. Perguntas na Avaliação Semanal

Grupo	Perguntas
Expectativa	Q1- Estou confiante que aprenderei o conteúdo e terei sucesso na disciplina.
	Q4- Estou entendendo e conseguindo aprender o conteúdo.
Valor	Q2- Estou achando as aulas da disciplina úteis para o curso.
	Q5- Estou entendendo a importância dessa disciplina no curso.
Custo	Q3- A disciplina exige muito tempo e estou tendo de me abdicar muito de outras coisas.
	Q6- Não estou conseguindo tempo necessário para me dedicar a essa disciplina.

O índice de motivação é calculado como:

$$EVC_{Total} = \sum EXP + VAL - CUS \qquad \text{INDICE}_{EVC} = \frac{EVC_{Total}}{\Delta EVC_{Total}}$$

Para responder a questão 3, foi analisado o índice de aprovação da disciplina e comparado aos semestres anteriores.

3. Resultados

A Tabela 2 apresenta os resultados das 8 avaliações, considerando somente os alunos que estiverem presentes na avaliação, pois a instituição registra como nota zero

⁶ <http://bit.ly/2xFpqty>

mesmo aqueles alunos que não fizeram a prova. As médias foram baixas, porém existe uma grande variação nas notas.

Tabela 2. Resultados das avaliações

Avaliações	Média (Desvio Padrão)	Qtde alunos
PC1	9,0 ($\pm 2,3$)	47
PC2	6,9 ($\pm 3,1$)	47
PC3	4,6 ($\pm 3,5$)	46
J1	3,4 ($\pm 3,1$)	44
J2	2,9 ($\pm 3,4$)	41
J3	4,6 ($\pm 3,6$)	37
J4	4,1 ($\pm 4,2$)	34
J5	2,4 ($\pm 3,0$)	26

A seguir são descritos e detalhados os resultados correspondentes a cada pergunta de análise.

Q1 - A participação nas missões do jogo tem relação como desempenho nas avaliações?

No jogo NoBug's Snack Bar é possível fazer o rastreamento e acompanhamento da quantidade de missões que cada aluno cumpriu. A Figura 2 apresenta a quantidade de alunos que concluíram cada uma das missões no jogo. As fases 2 a 4 (missões de 1 a 19) abrangem o assunto tratado da PC1, as fases 5 a 7 (missões 20 a 42) da PC2 e as fases 8 a 10 (missões 43 a 65) da PC3. Metade da turma cumpriu as fases 2 a 4, porém menos de 10% dos alunos concluíram o jogo.

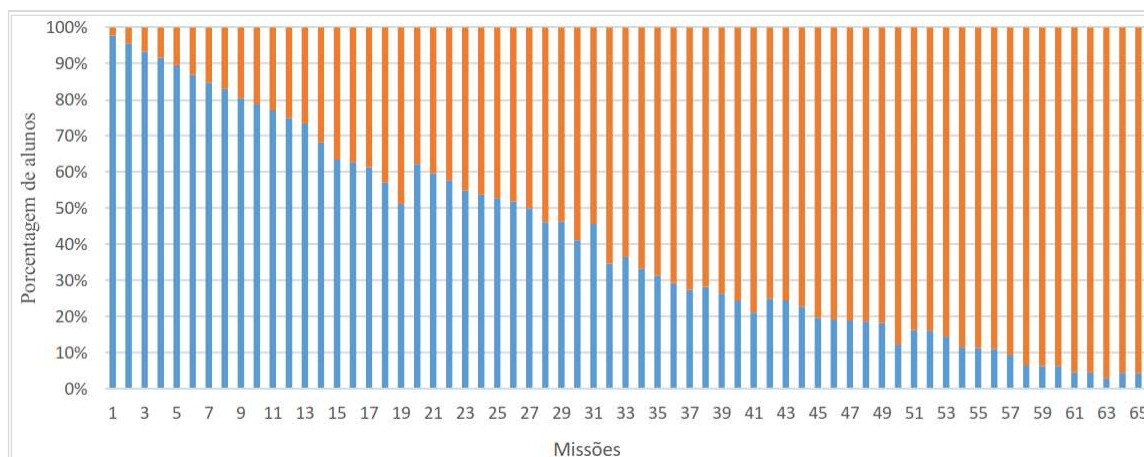


Figura 2. Jogadores que concluíram as missões

Para verificar se existe alguma relação entre as quantidades de missões resolvidas com as notas das avaliações, foi realizado um teste de correlação de Pearson com o resultado demonstrado na Tabela 3. Foram encontradas correlações fortes positivas indicando que a nota das provas está fortemente ligada com o engajamento do aluno em resolver as tarefas no jogo, inclusive na avaliação PC3, que não possui tarefas diretamente ligadas ao jogo.

Tabela 3. Correlação entre o desempenho nas provas e missões cumpridas no jogo

	PC1	PC2	PC3	Fases 2-4	Fases 5-7	Fases 8-10
PC1	-	0,710	0,688	0,763	0,722	0,612

PC2		-	0,718	0,741	0,766	0,705
PC3			-	0,697	0,752	0,748
Fases 2-4				-	0,860	0,811
Fases 5-7					-	0,834

Q2-A motivação na disciplina está relacionada ao resultado das avaliações?

A Figura 3 apresenta a motivação dos alunos atualizada em cada momento de avaliação. Percebe-se que o valor (importância) da disciplina teve pequena oscilação. Já a expectativa teve uma queda significativa após a avaliação J1, podendo indicar a dificuldade no entendimento do conteúdo pelos alunos ou reflexo do próprio desempenho ruim nas avaliações. A partir da avaliação PC3, percebe-se um aumento significativo na percepção do custo (tempo de dedicação necessário para a disciplina), podendo indicar novamente a dificuldade no entendimento quando se iniciou o uso de linguagem de programação Java, ou também o aumento no número e complexidade das atividades solicitadas.

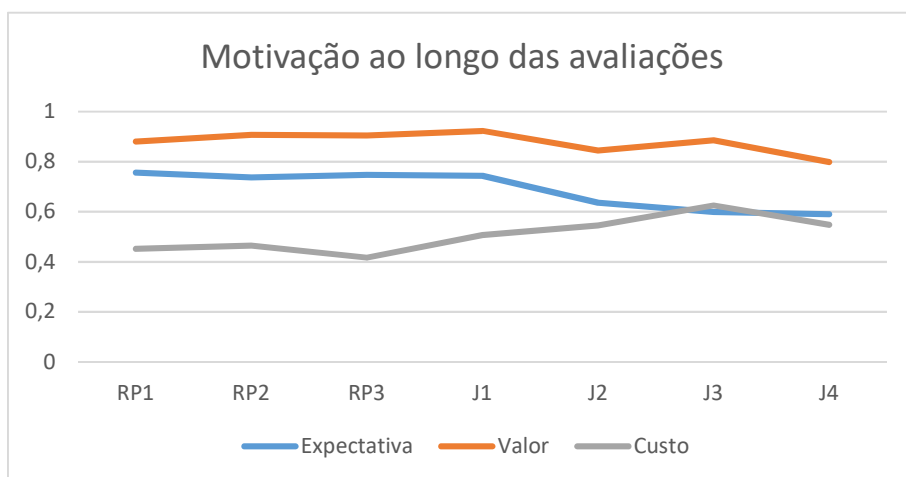


Figura 3. Evolução semanal da motivação

Considerando a motivação em geral (contemplando os 3 componentes: expectativa, valor e custo), a Figura 4 mostra a evolução da motivação a cada aplicação e os momentos das avaliações. Observa-se que, após a PC3, houve um acréscimo na motivação, porém um decréscimo acentuado já antes da própria prova J1, coincidindo com as baixas notas nessa avaliação. É importante observar que não foram realizadas coletas em todas as semanas, devido a eventos na universidade e atividades da disciplina.

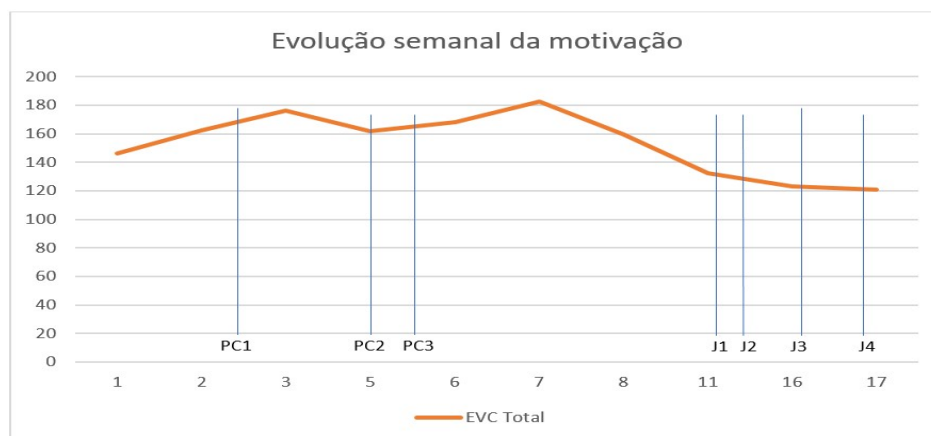


Figura 4. Evolução total semanal da motivação em relação às avaliações

Foi realizado um teste de correlação de Pearson entre as avaliações e a medição semanal das motivações (Tabela 4). Foi identificada uma correlação fraca entre a primeira avaliação e o índice de motivação até então. Após isso, a correlação torna-se moderada, aumentando até a avaliação J4, quando a correlação chega a 0.679. Curiosamente, a motivação nas primeiras semanas está correlacionada com as avaliações mais para o final do semestre.

Tabela 4. Correlação entre as avaliações e a motivação

	PC1	PC2	PC3	J1	J2	J3	J4	J5
M1	0,367	0,436	0,487	0,651	0,651	0,606	0,725	0,623
M2	0,396	0,471	0,521	0,658	0,647	0,586	0,733	0,566
M3	0,389	0,489	0,527	0,678	0,623	0,588	0,701	0,566
M4	0,382	0,492	0,459	0,646	0,595	0,604	0,630	0,589
M5	0,382	0,492	0,459	0,646	0,595	0,604	0,630	0,589
M6	0,388	0,496	0,471	0,670	0,604	0,650	0,669	0,568
M7	0,362	0,476	0,445	0,695	0,593	0,689	0,679	0,562

Considerando a escala EVC, o valor total da escala ($p=0.0000153$), expectativa ($p=0.0000588$) e custo ($p=0.00127$) mostraram ter variância significativa no resultado. Basicamente, estudantes aprovados mostraram maior motivação. Quanto ao custo, estudantes reprovados indicaram maiores valores para essa escala.

Q3 - O aumento no número de avaliações melhorou o aproveitamento dos estudantes?

Para avaliar a evolução da aprendizagem na primeira fase da disciplina (Momento 1), foi realizado um pré-teste no primeiro dia de aula e o mesmo teste foi repetido no final da sexta semana (PC3). A Figura 5 apresenta os resultados dos 33 alunos que fizeram o pré-teste. A maioria dos estudantes teve um acréscimo de desempenho, porém, para os alunos 11 e 14, o desempenho foi pior no pós-teste, e o aluno 30 zerou em ambos os testes. Foi executado um teste de hipótese *t-student* pareado unilateral à esquerda, com nível de significância de $\alpha=0,05$. Os resultados do teste ($T_{calc} = 9,574898 \cdot 10^{-10}$ e $T_{tab} = 2,0369$) demonstraram que $T_{calc} < T_{tab}$, considerando então que existem diferenças significativas entre as notas do pré e pós-teste. Apesar da média geral da PC3 ser baixa (4,6), pode-se concluir que houve um acréscimo na apropriação do conhecimento.

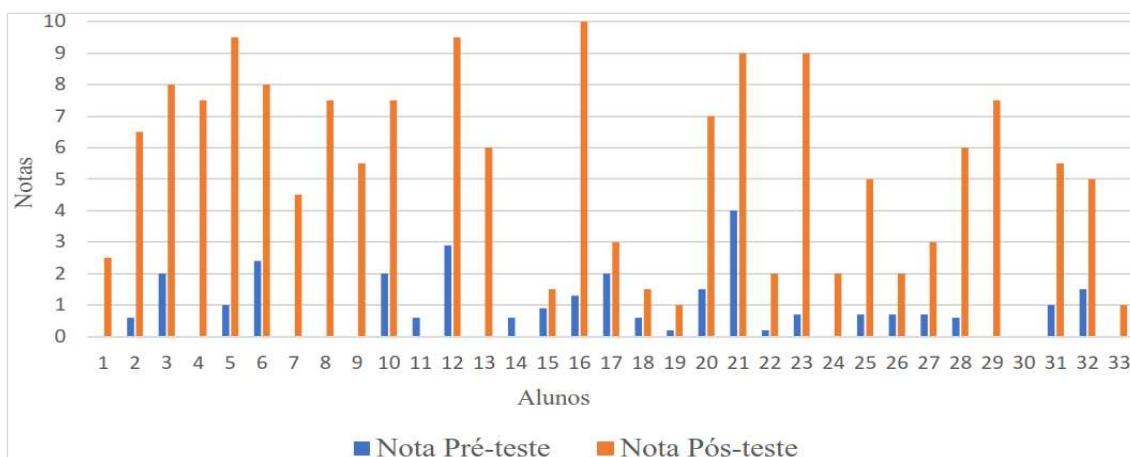


Figura 5. Notas do pré e pós-teste

Avaliando o conjunto das avaliações realizadas (Tabela 3), supõe-se que, apesar dos alunos terem se apropriado das habilidades do Pensamento Computacional, existiu algum problema quanto a transferência do conhecimento para a linguagem de programação. A Figura 6 mostra que o índice de aprovação nessa turma ficou significativamente abaixo da média histórica. A disciplina encerrou com 27% (n=13) de alunos aprovados, 20,8% (n=10) alunos reprovados por falta e 52,2% (n=25) reprovados por nota, o que pode demonstrar que aumentar a quantidade de avaliações, mesmo com menos assunto, não garantiu um melhor desempenho aos alunos.

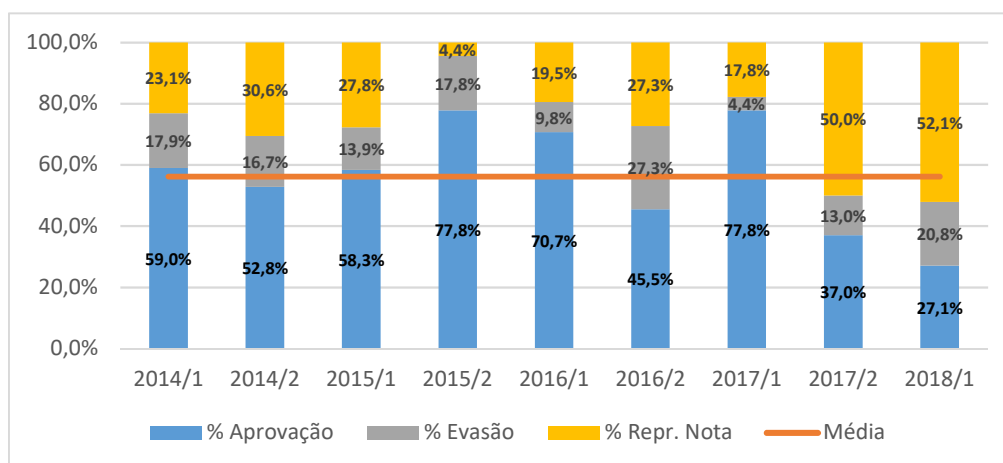


Figura 6. Resultados históricos de aprovação da disciplina

5. Considerações Finais

Esse trabalho apresentou o relato de um semestre da disciplina de Introdução à Programação com duração de 18 semanas tendo sido aplicadas 8 avaliações. A disciplina foi organizada em três momentos: o primeiro para o ensino de Pensamento Computacional, o segundo e terceiro para o ensino de programação em Java. Durante o período foram realizadas sete medições referentes à motivação dos alunos com a disciplina.

No primeiro momento da disciplina utilizou-se um jogo sério que permitiu verificar a quantidade de prática que os alunos realizaram. Como esperado, foram

encontradas correlações fortes positivas entre o desempenho nas avaliações e o trabalho dispendido pelos alunos no jogo (Q1). Nos momentos seguintes os alunos resolviam listas de exercícios em Java em seus computadores, não sendo monitorados, e por essa razão não se consegue medir o esforço real dispendido nesse período, mas percebeu-se uma redução tanto no desempenho quanto na motivação dos estudantes.

Foi identificado também uma correlação média da motivação com o desempenho nas avaliações e uma relação da motivação com o resultado final (Q2). De acordo com a motivação medida durante o semestre, os alunos apresentaram um aumento na percepção de esforço necessário para acompanhar a disciplina e, diminuíram a expectativa em serem aprovados, apesar de continuarem a concordarem na importância da disciplina na sua formação. Esses resultados indicam uma possível dificuldade no entendimento do conteúdo na migração do conteúdo de pensamento computacional para a linguagem de programação Java e/ou aumento no número e complexidade das atividades solicitadas, gerando uma sobrecarga para os alunos.

Devido ao baixo índice de aprovação (27%) em relação à média história dessa disciplina no curso, é possível determinar que o objetivo de melhorar o desempenho dos alunos com uma maior frequência de avaliações, não foi alcançado (Q3). Uma possível causa para isso é que o maior número de avaliações gera uma maior necessidade de tempo para estudo e, como as atividades (listas de exercícios) não foram monitoradas ao longo do tempo (ao contrário do primeiro momento), o *feedback* constante não ocorria para indicar o que o estudante deveria melhorar. Uma alternativa a ser avaliada é a realização de exercícios monitorados e com *feedback* constante aos alunos, também para atividades com a linguagem de programação Java. Ainda, noutra direção, é encontrar elementos motivadores para que eles mantenham foco nos estudos.

Como trabalho futuro, espera-se desenvolver um modelo de predição de aprovação, usando alguns dados como o desempenho escolar, a motivação inicial e semanal, entre outros, para que sirva de alerta tanto ao aluno quanto monitoramento para o professor.

Referências

- Cristina, V., & Aureliano, O. (2012). Ensino-aprendizagem de Programação para Iniciantes : uma Revisão Sistemática da Literatura focada no SBIE e WIE. In *Simpósio Brasileiro de Informática na Educação* (pp. 26–30).
- Eccles, J., Adler, T. F., Futterman, R., Goff, S. B., Kaczala, C. M., & Meece, J. (1983). Expectancies, values and academic behaviors. *Achievement and Achievement Motives: Psychological and Sociological Approaches*, 75–146.
- Gomes, A., & Mendes, A. J. (2007). Learning to program-difficulties and solutions. In *International Conference on Engineering Education* (pp. 1–5). Coimbra, Portugal.
- Gomes, M., Becker, L., Gestaro, L., Amaral, É., & Tarouco, L. (2015). Um estudo sobre erros em Programação : reconhecendo as dificuldades de programadores iniciantes. In *Workshops do IV Congresso Brasileiro de Informática na Educação* (pp. 1398–1407).
- Kosovich, J. J., Hulleman, C. S., Barron, K. E., & Getty, S. (2015). A Practical Measure of Student Motivation : Establishing Validity Evidence for the Scale in Middle School. *Journal of Early Adolescence*, 35(5–6), 790–816.

- Menezes, C. S. De, Alcina, I., & Nobre, M. (2002). Um Ambiente Cooperativo para Apoio a Cursos de Introdução a Programação. In *Workshop sobre Educação em Computação*.
- Moreira, B. G. (2014). Desenvolvimento de uma ferramenta de avaliação por pares para disciplinas de algoritmos e programação. In *Workshops do III Congresso Brasileiro de Informática na Educação* (pp. 68–75).
- Paes, R. D. B., Malaquias, R., & Guimarães, M. (2013). Ferramenta para a Avaliação de Aprendizado de Alunos em Programação de Computadores. In *Workshops do II Congresso Brasileiro de Informática na Educação* (pp. 203–212).
- Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., ... Paterson, J. (2007). A survey of literature on the teaching of introductory programming. In *12th Annual Conference on Innovation and Technology in Computer Science Education* (pp. 204–223). Dundee, Scotland.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137–172.
- Vahldick, A., Marcelino, M. J., & Mendes, A. J. (2017). Principles of a casual serious game to support introductory programming learning in higher education. In *Gamification-Based E-Learning Strategies for Computer Programming Education* (pp. 53–78). IGI Global.
- Vahldick, A., & Mattos, M. M. (2009). Aprendendo Programação de Computadores com Experiências Lúdicas. In *IV Internacional Conference on Engineering and Computer Education*. Buenos Aires, Argentina.
- Vihavainen, A., Luukkainen, M., & Kurhila, J. (2012). Multi-faceted Support for MOOC in Programming. In *13th Annual Conference on Information Technology Education* (Vol. 68).